REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
9 May 96

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Design and Application of Robust Reduced-Order Hybrid Position and Force Controllers for a Two-Link Flexible Manipulator

5. FUNDING NUMBERS

6. AUTHOR(S)

David E. Bossert

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFIT Student Attending:

University of Washington

8. PERFORMING ORGANIZATION REPORT NUMBER

96-016D

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DEPARTMENT OF THE AIR FORCE
AFIT/CI
2950 P STEET, BLDG 125
WRIGHT-PATTERSON AFB OH 45433-7765

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release IAW 190-1
Distribution Unlimited
BRIAN D. GAUTHIER, MSgt, USAF
Chief Administration

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

19960809 062

14. SUBJECT TERMS

15. NUMBER OF PAGES
233

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C -	Contract	**PR** -	Project
G -	Grant	**TA** -	Task
PE -	Program Element	**WU** -	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD -	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE -	See authorities.
NASA -	See Handbook NHB 2200.2.
NTIS -	Leave blank.

Block 12b. Distribution Code.

DOD -	Leave blank.
DOE -	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA -	Leave blank.
NTIS -	Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Design and Application of Robust Reduced-Order Hybrid Position and Force Controllers for a Two-Link Flexible Manipulator

by David E. Bossert
Captain, USAF

1996

233 Pages

Doctor Of Philosophy
Aeronautics and Astronautics

University Of Washington

This research focuses on the design and application of robust reduced-order hybrid position and force controllers on a two-link flexible manipulator. The first part of the research examines the design aspect of the hybrid controllers. Traditionally, full-order controllers are used in common techniques such as Linear-Quadratic-Gaussian (LQG) control. Full-order controllers have the disadvantages of requiring large amounts of memory as well as using excessive computational time. In addition, LQG-class controllers are only guaranteed to work at the design condition. With the use of the direct-optimization tool SANDY, reduced-order controllers are designed directly for several plant conditions (robustness). The experimental results show that the robust reduced-order SANDY controllers provide comparable performance to the

full-order LQG controllers in the applications of position control, force control, and hybrid position and force control.

The second thrust of the research is on the application of these controllers. In a typical manufacturing task, there are several stages involved. This work develops a framework which allows a tool to move to an unknown surface whose exact location is not known, perform a hybrid position and force control task on the surface, and return to the surface. The technique performs well experimentally for both full-order and reduced-order controllers on an unknown straight line surface and an unknown arc surface. Also, the structure could easily be adapted to the case where a surface shape is known exactly. The approach shows promise in many manufacturing applications.

Bibliography

1 Kraft, R., "Experiments in End-Point Control of a Flexible Robot with a Mini-Manipulator", Ph.D. Dissertation, Department of Aeronautics and Astronautics, Stanford University, May 1989.

2 Ly, U., "A Design Algorithm for Robust Low-Order Controller," Ph.D. Dissertation, Department of Aeronautics and Astronautics, Stanford University, Nov 1982.

3 Oakley, C., "Experiments in Modeling and End-Point Control of Two-Link Flexible Manipulators", Ph.D. Dissertation, Stanford University, 1991.

4 Raibert, M., and Craig, J., "Hybrid Position/Force Control of Manipulators", *J. of Dynamic Systems, Measurement, and Control*, 102, pp. 126-133, 1981.

Design and Application of Robust Reduced-Order Hybrid Position and Force Controllers for a Two-Link Flexible Manipulator

by David E. Bossert
Captain, USAF

1996

233 Pages

Doctor Of Philosophy
Aeronautics and Astronautics

University Of Washington

This research focuses on the design and application of robust reduced-order hybrid position and force controllers on a two-link flexible manipulator. The first part of the research examines the design aspect of the hybrid controllers. Traditionally, full-order controllers are used in common techniques such as Linear-Quadratic-Gaussian (LQG) control. Full-order controllers have the disadvantages of requiring large amounts of memory as well as using excessive computational time. In addition, LQG-class controllers are only guaranteed to work at the design condition. With the use of the direct-optimization tool SANDY, reduced-order controllers are designed directly for several plant conditions (robustness). The experimental results show that the robust reduced-order SANDY controllers provide comparable performance to the

full-order LQG controllers in the applications of position control, force control, and hybrid position and force control.

The second thrust of the research is on the application of these controllers. In a typical manufacturing task, there are several stages involved. This work develops a framework which allows a tool to move to an unknown surface whose exact location is not known, perform a hybrid position and force control task on the surface, and return to the surface. The technique performs well experimentally for both full-order and reduced-order controllers on an unknown straight line surface and an unknown arc surface. Also, the structure could easily be adapted to the case where a surface shape is known exactly. The approach shows promise in many manufacturing applications.

Bibliography

1 Kraft, R., "Experiments in End-Point Control of a Flexible Robot with a Mini-Manipulator", Ph.D. Dissertation, Department of Aeronautics and Astronautics, Stanford University, May 1989.

2 Ly, U., "A Design Algorithm for Robust Low-Order Controller," Ph.D. Dissertation, Department of Aeronautics and Astronautics, Stanford University, Nov 1982.

3 Oakley, C., "Experiments in Modeling and End-Point Control of Two-Link Flexible Manipulators", Ph.D. Dissertation, Stanford University, 1991.

4 Raibert, M., and Craig, J., "Hybrid Position/Force Control of Manipulators", *J. of Dynamic Systems, Measurement, and Control*, 102, pp. 126-133, 1981.

Design and Application of Robust Reduced-Order Hybrid Position and Force Control for a Two-Link Flexible Manipulator

by

David Edward Bossert

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

1996

Approved by ________________________________
Chairperson of Supervisory Committee

Program Authorized
to Offer Degree: Aeronautics and Astronautics

Date 5/10/96

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to University Microfilms, 1490 Eisenhower Place, P.O. Box 975, Ann Arbor, MI 48106, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform.

Signature ______________

Date 9 May 96

University of Washington

Abstract

Design and Application of Robust Reduced-Order Hybrid Position and Force Controllers for a Two-Link Flexible Manipulator

by David E. Bossert

Chairperson of the Supervisory Committee: Professor Juris Vagners
Department of Aeronautics and Astronautics

This research focuses on the design and application of robust reduced-order hybrid position and force controllers on a two-link flexible manipulator. The first part of the research examines the design aspect of the hybrid controllers. Traditionally, full-order controllers are used in common techniques such as Linear-Quadratic-Gaussian (LQG) control. If the plant is of large order, then the resulting controller is also of large order. Full-order controllers have the disadvantages of requiring large amounts of memory as well as using excessive computational time. In addition, LQG-class controllers are only guaranteed to work at the design condition. With the use of the direct-optimization tool SANDY, reduced-order controllers are designed directly for several plant conditions (robustness). The experimental results show that the robust reduced-order SANDY controllers provide comparable performance to the full-order LQG controllers in the applications of position control, force control, and

hybrid position and force control.

The second thrust of the research is on the application of these controllers. In a typical manufacturing task, there are several stages involved. First, the tool must move from some initial starting position and make contact with a surface, and then it must perform some contact task (constrained motion), and finally return to its initial location to restart the procedure on the next item. This work develops a framework which allows a tool to move to an unknown surface whose exact location is not known, perform a hybrid position and force control task on the surface, and then retract to the starting position. The complete (unified) hybrid task method is presented as a 6-phase process, and is experimentally evaluated on a two-link manipulator. The technique performs well experimentally for both full-order and reduced-order controllers on an unknown straight line surface and an unknown arc surface. Also, the structure could easily be adapted to the case where a surface shape is known exactly. The approach shows promise in many manufacturing applications.

Table of Contents

List of Figures iv
List of Tables x
List of Abbreviations and Acronyms xi
List of Symbols xiii
Chapter 1: Introduction 1
1.1 Motivation 1
1.2 Contributions 2
1.3 Organization 3
Chapter 2: Literature Review 5
2.1 Current Flexible Manipulator Facilities 5
2.2 Current Flexible Manipulator Applications 5
2.3 Constrained Motion Control 6
2.4 Surface Following Applications 7
2.5 The Complete (Unified) Hybrid Task 8
2.6 Summary 8
Chapter 3: Experimental Setup 9
3.1 General Description 9
3.2 Actuators 9
3.3 Links 10
3.4 Sensors 11
3.5 Data Acquisition System 13
3.6 Software Interface 14
Chapter 4: Position Dynamics Model 17
4.1 Non-Linear Equations of Motion 17
4.2 Linearized Equations of Motion 19
4.3 End-point Calculations 20
4.4 Low-Torque Model 20
4.5 High-Torque Model 24
4.6 Model Reduction 24
4.7 Experimental Model Validation 30
4.8 Mode Shapes 34
Chapter 5: PID Design and Experiments 36
5.1 PID Design 36
5.2 PID Control Bandwidth 39
Chapter 6: LQG-Class Controllers Design and Experiments 41
6.1 LQG/LTR 46
6.2 Special Control Basis Loop Transfer Recovery (SCB/LTR) 48
6.3 Controller Reduction 55
6.4 Control Bandwidths 58
6.5 Observations 59
Chapter 7: SANDY Background 61
Chapter 8: SANDY Double-Angle Control Design 64
8.1 SANDY Control Bandwidth 67

8.2 Experimental Comparison of SANDY and PID Controllers 69
Chapter 9: Sandy Robust Design and Endpoint Position Test 73
9.1 Inverse Kinematics 74
9.2 TLA Application 76
9.3 Endpoint Trajectory Generation 78
9.3.1 Arc 80
9.3.2 Sine Wave 82
9.4 Robust SANDY Design 83
9.5 Experimental Comparison 87
Chapter 10: Hybrid Position and Force Control Overview 96
10.1 Force Control 96
10.1.1 Jacobian 96
10.1.2 Principle of Virtual Work 98
10.1.3 Identification of the Force Transfer Functions 99
10.1.4 Force Components and Coordinates 100
10.1.5 Force Control Structure 102
10.2 Position Control 103
10.2.1 Position Control With Surface Knowledge 104
10.2.2 Position Control Without Surface Knowledge 104
10.3 Hybrid Control Structure 106
10.4 Task Overview 107
Chapter 11: Force Control Model and Design 110
11.1 Force Models 110
11.2 Force Control Designs 111
11.2.1 PI Force Control 111
11.2.2 LQG Force Control 111
11.2.3 LQG Reduced-Order Force Control 115
11.2.4 SANDY Force Control Design 116
11.3 Experimental Force Controller Comparison 117
Chapter 12: Hybrid Control Designs 120
12.1 Superposition 120
12.2 Complete Optimization 128
12.2.1 Complete Reduced-Order Optimization 128
12.2.2 Complete Robust Reduced-Order Optimization 133
12.3 Hybrid Design Experimental Conclusions 137
Chapter 13: Surface Following Experiments 138
13.1 Surface Shape Effects 138
13.2 Effect of Tracking Speed Reference 141
13.3 Effect of Commanded Force 143
13.4 Effect of Different Controllers 144
13.5 Modified Tangent Speed Command 145
13.6 Surface Following Conclusions 149
Chapter 14: Surface Touchdown 151

14.1 Baseline Touchdown Results 152
14.2 Touchdown Control Strategies 156
14.3 Experimental Implementation 157
Chapter 15: The Complete Hybrid Task 158
Chapter 16: Conclusion 168
16.1 Summary 168
16.2 Design Lessons Learned 168
16.2.1 Hardware Variability 169
16.2.2 Optimization Observations 169
16.2.3 Logical Design Steps 170
16.3 Recommendations for Future Research 171
Appendix Q: Arm Modeling Details 176
Q.1 Assumptions 176
Q.2 Parameters 176
Q.3 Nonlinear Equations of Motion 178
Q.4 Linearized Two-Link Equations of Motion 178
Q.5 Non-Linear Two-Link 6 DOF Dynamic Equations 179
Q.6 Closed Form Linearized Equations of Motion 188
Appendix R: - Additional SANDY Double Angle and PID Plots 191
R.1 SANDY Double Angle Plots 194
R.2 SANDY vs. PID Plots 196
Appendix S: - Additional LQG-Class Plots 198
S.1 LTR Plots 200
S.2 7th Order LTR Plots 204
S.3 Overall Comparison Plots 206
Appendix T: - Additional Endpoint Tracking Plots 209
Appendix U: - MATLAB m.files 213
U.1 Model Generation m.files 213
U.2 SANDY Design m.files 226
U.3 General Design 233

List of Figures

Figure 3.1- Block Diagram of TLA Hardware 9
Figure 3.2- TLA Facility Picture 10
Figure 3.3- Actuator Conversion Factors 11
Figure 3.4- ATI Force Sensor Block Diagram 12
Figure 3.5- Force Sensing Mechanism 13
Figure 3.6- Force/Torque Conventions 13
Figure 3.7- System Build Simulation Block 14
Figure 3.8- Interactive Animation Block Diagram 15
Figure 3.9- Simulation Plant Implementation 16
Figure 3.10- Typical System Build Controller Structure 16
Figure 4.1- Lumped-Spring-Mass-Damper Model Parameters 18
Figure 4.2- Low-Torque Model Bode Plot, θ_1/τ_1 23
Figure 4.3- Low-Torque Model Bode Plot, θ_2/τ_2 24
Figure 4.4- High-Torque Model Bode Plot, θ_1/τ_1 26
Figure 4.5- High-Torque Model Bode Plot, θ_2/τ_2 26
Figure 4.6- High-Torque Model Bode Plot, θ1/τ1 of the Reduced-Order Model 29
Figure 4.7- High-Torque Model Bode Plot, θ2/τ2 of the Reduced-Order Model 29
Figure 4.8- Low -Torque Model Bode Plot, θ1/τ1 of the Reduced-Order Model 30
Figure 4.9- Low -Torque Model Bode Plot, θ2/τ2 of the Reduced-Order Model 30
Figure 4.10- Experimental Sine Sweep Block Diagram (Link 2) 31
Figure 4.11- Shoulder Magnitude Frequency Response 32
Figure 4.12- Shoulder Phase Frequency Response 32
Figure 4.13- Elbow Magnitude Frequency Response 33
Figure 4.14- Elbow Phase Frequency Response 33
Figure 4.15- 1st Mode, Semi-Flexible Links (Link 1, Mode 1) 34
Figure 4.16- 2nd Mode, Semi-Flexible Links (Link 2, Mode 1) 35
Figure 4.17- 3rd Mode, Semi-Flexible Links (Link 1, Mode 2) 35
Figure 5.1- TLA Test Trajectories 36
Figure 5.2- Trajectory 2 Reference Commands 37
Figure 5.3- Angle Conventions 37
Figure 5.4- PID Experimental Results, Trajectory 2 38
Figure 5.5- PID Experimental vs. Simulation, Trajectory 2, θ_1 ..38
Figure 5.6- PID Experimental vs. Simulation, Trajectory 2, θ_2 ..39
Figure 5.7- PID θ_1/θ_{1cmd} Control Bandwidth 39
Figure 5.8- PID θ_2/θ_{2cmd} Control Bandwidth 40
Figure 6.1- LQG Controller Experimental Run - Traj 2 46
Figure 6.2- LQG, Exp vs. Sim, Traj 2, θ_1 47
Figure 6.3- LQG, Exp vs. Sim, Traj 2, θ_2 47
Figure 6.4- LQG/LTR Controller Experimental Run - Traj 2 49
Figure 6.5- LQR Target Loop 52
Figure 6.6- Output Feedback Loop 52
Figure 6.7- LTR Controller Experimental Run - Traj 2 55

Figure 6.8- LQG/LTR 7th Order Reduced Controller Exp Response- Traj 2 . . .56
Figure 6.9- LQG/LTR 7th Order Reduced Controller Exp vs. Sim, θ_1 - Traj 2 .57
Figure 6.10- LQG/LTR 7th Order Reduced Controller Exp vs. Sim, θ_2- Traj 2 .57
Figure 6.11- SCB/LTR 7th Order Reduced Controller Exp Response- Traj 2 . .58
Figure 6.12- θ_1/θ_{1cmd} LQG Control Bandwidth 59
Figure 6.13- θ_2/θ_{2cmd} LQG Control Bandwidth 59
Figure 7.1- Sandy Feedback Control Structure 62
Figure 8.1- SANDY Experimental Run, Trajectory 2 65
Figure 8.2- SANDY Experimental vs. Simulation, Trajectory 2, θ_1 66
Figure 8.3- SANDY Experimental vs. Simulation, Trajectory 2, θ_2 66
Figure 8.4- Control Bandwidth for the Shoulder Motor 67
Figure 8.5- Control Bandwidth for the Elbow Motor 68
Figure 8.6- SANDY Bandwidth With Prefilter, Shoulder Motor 68
Figure 8.7- SANDY Bandwidth With Prefilter, Elbow Motor 69
Figure 8.8- PID vs. SANDY, Traj 2, θ_1 70
Figure 8.9- PID vs. SANDY, Traj 2, θ_2 70
Figure 8.10- Robust PID vs. SANDY, Traj 1, θ_1 71
Figure 8.11- Robust PID vs. SANDY, Traj 1, θ_2 71
Figure 9.1- Forward Kinematic Conventions 73
Figure 9.2- TLA Physical Angle Redundancy 74
Figure 9.3- Inverse Kinematic Angle Conventions 75
Figure 9.4- TLA Endpoint Trajectory Implementation .77
Figure 9.5- 0.21m Straight Line Reference Trajectory .78
Figure 9.6- Hub Angle Spline Input Reference Trajectory 79
Figure 9.7- Endpoint Spline Reference Trajectory 79
Figure 9.8- 40° Arc Reference Trajectory 80
Figure 9.9- Graphical Depiction of the Arc Trajectory .81
Figure 9.10- Sine Wave Reference Trajectory 82
Figure 9.11- Graphical Depiction of the Sine Trajectory 83
Figure 9.12- Filtered Shoulder Bandwidth, Robust 4th order SANDY 86
Figure 9.13- Filtered Elbow Bandwidth, Robust 4th order SANDY 87
Figure 9.14- Exp 3 Sec Line: LQG, SANDY, Ref from x,y=.64,.61 - .99,.25 m. 88
Figure 9.15- Exp 10 Sec Line: LQG, SANDY, Ref from x,y=.64,.61 - .99.25 m 88
Figure 9.16- Exp 3 Sec Arc: LQG, SANDY, Ref for R=.51", θ=40 deg 89
Figure 9.17- Exp 10 Sec Arc: LQG, SANDY, Ref for R=.51", θ=40 deg 89
Figure 9.18- Exp 6 Sec Sine: LQG, SANDY, Ref for Amp=.1m, Length=.25m 90
Figure 9.19- Exp 20 Sec Sine: LQG, SANDY, Ref for Amp=.1m, Length=.25m 91
Figure 9.20- Exp 10 Sec Line: LQG, SANDY, Ref from x,y=.64,.61 -
.99.25 m, Light Tip 91
Figure 9.21- Exp 10 Sec Arc: LQG, SANDY, Ref for R=.51", θ=40 deg,
Light Tip 92
Figure 9.22- Exp 20 Sec Sine: LQG, SANDY, Ref for Amp=.1m,
Length=.25m, Light Tip 92
Figure 9.23- Exp 10 Sec Line: LQG, SANDY, Ref from x,y=.64,.61 -

.99.25 m, Heavy Tip 93
Figure 9.24- Exp 10 Sec Arc: LQG, SANDY, Ref for R=.51", θ=40 deg, Heavy Tip 94
Figure 9.25- Exp 20 Sec Sine: LQG, SANDY, Ref for Amp=.1m, Length=.25 m, Heavy Tip 94
Figure 10.1- Jacobian Parameters for the TLA 98
Figure 10.2- Identification by Approximate Linearization 99
Figure 10.3- Surface Force Diagram 100
Figure 10.4- Relationship Between Inertial and Measured Frames 101
Figure 10.5- Force Control Scheme 103
Figure 10.6- Surface Following Algorithm 104
Figure 10.7- Hybrid Force and Position Control Scheme 106
Figure 10.8- Surface Following Task 108
Figure 10.9- Available Surfaces 109
Figure 11.1- Experimental vs. Sim, PI=.3,5 , Fx Controller 112
Figure 11.2- Experimental vs. Sim, PI=.7,6 , Fy Controller 112
Figure 11.3- LQG Fx Controller Bandwidth 114
Figure 11.4- LQG Fy Controller Bandwidth 115
Figure 11.5- SANDY Fx Controller Bandwidth 117
Figure 11.6- SANDY Fy Controller Bandwidth 118
Figure 11.7- Experimental Controller Fx Comparison 119
Figure 11.8- Experimental Controller Fy Comparison 119
Figure 12.1- 6th Order Robust SANDY Shoulder Control Bandwidth 121
Figure 12.2- 6th Order Robust SANDY Elbow Control Bandwidth 122
Figure 12.3- LQG Position and LQG Force End-Point History 124
Figure 12.4- LQG Position and LQG Force Experimental Data 124
Figure 12.5- LQG Position and LQGR Force Experimental Data 125
Figure 12.6- LQG Position and SANDY Force Experimental Data 126
Figure 12.7- SANDY Position and LQG Force Experimental Data 126
Figure 12.8- SANDY Position and LQGR Force Experimental Data 127
Figure 12.9- SANDY Position and Force Experimental Data 127
Figure 12.10- Shoulder Bandwidth, SANDY Hybrid Position Controller 131
Figure 12.11- Elbow Bandwidth, SANDY Hybrid Position Controller 131
Figure 12.12- Fx Control Bandwidth, SANDY Hybrid Controller 132
Figure 12.13- Fy Control Bandwidth, SANDY Hybrid Controller 132
Figure 12.14- Complete Hybrid Optimization Experimental Data 133
Figure 12.15- Shoulder Bandwidth, SANDY Robust Hybrid Controller 135
Figure 12.16- Elbow Bandwidth, SANDY Robust Hybrid Controller 135
Figure 12.17- Fx Bandwidth, SANDY Robust Hybrid Controller 136
Figure 12.18- Fy Bandwidth, SANDY Robust Hybrid Controller 136
Figure 12.19- Complete Robust Hybrid Optimization Experimental Data 137
Figure 13.1- LQG, Line, 50gf Command, 1 cm/s, Strip 139
Figure 13.2- LQG, Line, 50gf Command, 1 cm/s, x-y 139
Figure 13.3- LQG, Concave Arc, 50gf, 1 cm/s, Strip 140

Figure 13.4- LQG, Concave Arc, 50gf, 1 cm/s, x-y 140
Figure 13.5- LQG, Convex Arc, 50 gf, 1 cm/s, Strip 141
Figure 13.6- LQG, Convex Arc, 50 gf, 1 cm/s, x-y 142
Figure 13.7- LQG, Line, 50gf Command, 2 cm/s, Strip 142
Figure 13.8- LQG, Line, 50gf Command, 2 cm/s, x-y 143
Figure 13.9- LQG, Line, 30gf Command, 1 cm/s, Strip 143
Figure 13.10- LQG, Line, 30gf Command, 1 cm/s, x-y 144
Figure 13.11- SANDY, Line, 50gf Command, 1 cm/s, Strip 145
Figure 13.12- SANDY, Line, 50gf Command, 1 cm/s, x-y 145
Figure 13.13- LQG, Line, 50gf, Mod, 1 cm cmd, strip 146
Figure 13.14- LQG, Line, 50gf, Mod, 1 cm cmd, x-y 147
Figure 13.15- LQG, Concave Arc, 50gf, Mod, 1 cm cmd, strip 147
Figure 13.16- LQG, Concave Arc, 50gf, Mod, 1 cm cmd, x-y 148
Figure 13.17- SANDY, Line, 50gf, Mod, 1 cm cmd, strip 148
Figure 13.18- SANDY, Line, 50gf, Mod, 1 cm cmd, x-y 149
Figure 14.1- Baseline Hybrid Force and Position Control Scheme 151
Figure 14.2- Impact Force vs. Impact Speed 153
Figure 14.3- 0.6 cm/s Touchdown with 50 gf Reference, Baseline Performance 153
Figure 14.4- 1.6 cm/s Touchdown with 50 gf Reference, Baseline Performance 154
Figure 14.5- 0.8 cm/s Touchdown with 50 gf Reference, Baseline Performance 155
Figure 14.6- 0.8 cm/s Touchdown with 50 gf Reference, Surface Reference ...155
Figure 14.7- Modified Hybrid Force and Position Control Scheme 156
Figure 14.8- Experimental Results, Modified Touchdown Scheme 157
Figure 15.1- Complete Hybrid Task Description 158
Figure 15.2- LQG, Unknown Line Strip Chart 161
Figure 15.3- LQG, Unknown Line End-Point Position 162
Figure 15.4- SANDY, Unknown Line, Strip Chart 163
Figure 15.5- SANDY, Unknown Line, End-Point Position 164
Figure 15.6- LQG, Unknown Arc, Strip Chart 164
Figure 15.7- LQG, Unknown Arc, End-Point Position 165
Figure 15.8- SANDY, Unknown Arc, Strip Chart 165
Figure 15.9- SANDY, Unknown Arc, End-Point Position 166
Figure B.1- PID Experimental Run, Trajectory 1 191
Figure B.2- PID Experimental vs. Simulation, Trajectory 1, θ_1 192
Figure B.3- PID Experimental vs. Simulation, Trajectory 1, θ_2 192
Figure B.4- PID Experimental Run, Trajectory 3 193
Figure B.5- PID Experimental vs. Simulation, Trajectory 3, θ_1 193
Figure B.6- PID Experimental vs. Simulation, Trajectory 3, θ_2 193
Figure B.7- SANDY Experimental Run, Trajectory 1 194
Figure B.8- SANDY Experimental vs. Simulation, Trajectory 1, θ_1 194
Figure B.9- SANDY Experimental vs. Simulation, Trajectory 1, θ_2 195
Figure B.10- SANDY Experimental Run, Trajectory 3 195
Figure B.11- SANDY Experimental vs. Simulation, Trajectory 3, θ_1 195
Figure B.12- SANDY Experimental vs. Simulation, Trajectory 3, θ_2 196

Figure B.13- SANDY vs. PID Experimental, Trajectory 1, θ_1 196
Figure B.1-SANDY vs. PID Experimental, Trajectory 1, θ_2 196
Figure B.2- SANDY vs. PID Experimental, Trajectory 3, θ_1 197
Figure B.3- SANDY vs. PID Experimental, Trajectory 3, θ_2 197
Figure C.1- LQG/LTR Experimental Run, Trajectory 1 198
Figure C.2- LQG/LTR Experimental vs. Simulation, Trajectory 1, θ_1 198
Figure C.3- LQG/LTR Experimental vs. Simulation, Trajectory 1, θ_2 199
Figure C.4- LQG/LTR Experimental Run, Trajectory 3 199
Figure C.5- LQG/LTR Experimental vs. Simulation, Trajectory 3, θ_1 199
Figure C.6- LQG/LTR Experimental vs. Simulation, Trajectory 3, θ_2 200
Figure C.7- LTR Experimental Run, Trajectory 1 200
Figure C.8- LTR Experimental vs. Simulation, Trajectory 1, θ_1 200
Figure C.9- LTR Experimental vs. Simulation, Trajectory 1, θ_2 201
Figure C.10- LTR Experimental Run, Trajectory 3 201
Figure C.11- LTR Experimental vs. Simulation, Trajectory 3, θ_1 201
Figure C.12- LTR Experimental vs. Simulation, Trajectory 3, θ_2 202
Figure C.13-7th Order LQG/LTR Experimental Run, Trajectory 1 202
Figure C.14- 7th Order LQG/LTR Experimental vs. Sim, Trajectory 1, θ_1 202
Figure C.15- 7th Order LQG/LTR Experimental vs. Sim, Trajectory 1, θ_2 203
Figure C.16- 7th Order LQG/LTR Experimental Run, Trajectory 3 203
Figure C.17- 7th Order LQG/LTR Experimental vs. Simulation, Traj 3, θ_1 203
Figure C.18- 7th Order LQG/LTR Experimental vs. Simulation, Traj 3, θ_2 204
Figure C.19- 7th Order LTR Experimental Run, Trajectory 1 204
Figure C.20- 7th Order LTR Experimental vs. Simulation, Trajectory 1, θ_1 . . .204
Figure C.21- 7th Order LTR Experimental vs. Simulation, Trajectory 1, θ_2 . . .205
Figure C.22- 7th Order LTR Experimental Run, Trajectory 3 205
Figure C.23- 7th Order LTR Experimental vs. Simulation, Trajectory 3, θ_1 . . .205
Figure C.24- 7th Order LTR Experimental vs. Simulation, Trajectory 3, θ_2 . . .206
Figure C.25- Comparison Experimental Runs, Trajectory 1, θ_1 206
Figure C.26- Comparison Experimental Runs, Trajectory 1, θ_2 206
Figure C.27- Comparison Experimental Runs, Trajectory 2, θ_1 207
Figure C.28- Comparison Experimental Runs, Trajectory 2, θ_2 207
Figure C.29- Comparison Experimental Runs, Trajectory 3, θ_1 208
Figure C.30- Comparison Experimental Runs, Trajectory 3, θ_2 208
Figure D.1- Line Traj, 3 Sec Slew, Exp. vs. Sim. 209
Figure D.2- Line Traj, 10 Sec Slew, Exp. vs. Sim. 210
Figure D.3- Arc Traj, 3 Sec Slew, Exp. vs. Sim. 210
Figure D.4- Arc Traj, 10 Sec Slew, Exp. vs. Sim. 211
Figure D.5- Sine Traj, 6 Sec Slew, Exp. vs. Sim. 211
Figure D.6- Sine Traj, 20 Sec Slew, Exp. vs. Sim. 212

List of Tables

Number Page

Table 4.1 - PID Parameter ID gains 21
Table 4.2- Test Slews for Parameter ID 21
Table 4.3- Experimentally Tuned Values for the Low-Torque Model 21
Table 4.4 - Low-Torque Model Eigenvalues 22
Table 4.5- Experimentally Tuned Values for the High-Torque Model 24
Table 4.6- High-Torque Model Eigenvalues 25
Table 4.7- 18th Order Reduced High-Torque Model Eigenvalues 27
Table 4.8- 33rd Order Reduced Low-Torque Model Eigenvalues 28
Table 5.1- High Performance PID Gains 36
Table 5.2- Test Slews for Parameter ID 37
Table 6.1- LQR Optimal Gain G 43
Table 6.2- LQE Optimal K (LQG) 44
Table 6.3- LQG/LTR Optimal K (LQG) 48
Table 6.4- 18th Order Model Invariant Zeros 51
Table 6.5- LTR Optimal K_f 54
Table 9.1- Law of Cosine TLA Terms 75
Table 9.2- Some TLA Kinematic Relationships 76
Table 9.3- Robust Plant Conditions 84
Table 9.4- Optimization Plant Variations 84
Table 12.1- Hybrid Superposition Combinations 123
Table 12.2- Hybrid Pseudo-Plants 129
Table 12.3- Hybrid Optimization Weightings 130
Table 12.4- Robust Hybrid Optimization Weightings 134
Table 14.1- Factors Affecting Touchdown 156
Table A.1 -TLA Manipulator Parameters 176
Table A.2- LSMD Parameters 177

List of Abbreviations and Acronyms

AC100/C30 - ISI's real-time data acquisition and control system

Amp - amplitude

BW - control bandwidth

cm - centimeter

cmd - command

c.o.m. - center of mass

CCW - counter-clockwise

CPU - central processing unit

deg - degrees

F/T - force and torque

gf - grams-force

Hz - Hertz (cycles/second)

ISI - Integrated Systems Incorporated

LQE - Linear-Quadratic-Estimator

LQG - Linear-Quadratic-Gaussian

LQG/LTR - Linear-Quadratic-Gaussian with Loop Transfer Recovery

LSMD - lumped spring-mass-damper

LTR - loop transfer recovery

m - meter

Matlab - control tools software program

MATRIXx - control tools software program

Nm - Newton-meters

PI - proportional-integral

PID - proportional-integral-derivative

SANDY - direct optimization design tool

SCB - Special Coordinate Basis

SCB/LTR - Special Coordinate Basis Loop Transfer Recovery

SLA_A - Single flexible link with the tip bracket centered on the tip pad

SLA_B - Single flexible link with no tip load

TLA_A - Two flexible links with the elbow stator tied to the first link

TLA_B - Two flexible links with the elbow stator tied to the second link

TLA - Two-Link Flexible Manipulator (Two-Link Arm)

Traj - trajectory

V - Volts

List of Symbols

$\mathbf{A}$ - state matrix

$\mathbf{A}_{hyb}$ - hybrid controller state matrix

$\mathbf{A}_p$ - position controller state matrix

$\mathbf{A}_{fx}$ - x-force controller state matrix

$\mathbf{A}_{fy}$ - y-force controller state matrix

$\mathbf{B}$ - control input coupling matrix

$\mathbf{B}_{hyb}$ - hybrid controller control input coupling matrix

$\mathbf{B}_p$ - position controller control input coupling matrix

$\mathbf{B}_{fx}$ - x-force controller control input coupling matrix

$\mathbf{B}_{fy}$ - y-force controller control input coupling matrix

b_i - element torsional damping in link i

$\mathbf{C}$ - state output matrix

$\mathbf{C}_{hyb}$ - hybrid controller state output matrix

$\mathbf{C}_p$ - position controller state output matrix

$\mathbf{C}_{fx}$ - x-force controller state output matrix

$\mathbf{C}_{fy}$ - y-force controller state output matrix

$\mathbf{C}(s)$ - controller transfer function

c_{hi} - distance from shaft i to the c.o.m of hub i

c_{ti} - distance from motor shaft i + 1 to the center of mass of tip i

$\mathbf{D}$ - direct control output coupling matrix

$\mathbf{D}_{hyb}$ - direct hybrid controller sensor to control output coupling matrix

$\mathbf{D}_p$ - position controller output coupling matrix

$\mathbf{D}_{fx}$ - x-force controller output coupling matrix

$\mathbf{D}_{fy}$ - y-force controller output coupling matrix

d_i - length of link i

del_x_{ep} - local x Cartesian tip position

del_y_{ep} - local y Cartesian tip position

EI - stiffness

e_{fx} - x-force error

e_{fy} - y-force error

E[ ] - expectation operator

E(s) - loop error

Fx,y,z - Force in the x, y, or z direction

fxcom - x-force reference command

fxm - measured x-force

fxmc - x-force reference command in the measured reference frame

fxtf - x-force transfer function

fycom - y-force reference command

fym - measured y-force

fymc - y-force reference command in the measured reference frame

fytf - y-force transfer function

G - optimal control gain

Gi - optimal control gain (integral)

Ge - optimal control gain (estimator)

H - integral selection matrix

I_{hi} - Inertia of hub i about its center of mass

I_{ti} - Inertia of tip i about its center of mass

I_{li} - Inertia of link i

J - Quadratic performance index

J(X) - Jacobian

K - torsional stiffness matrix; optimal estimator gain matrix

k_p - proportional gain

k_I - integral gain

k_d - derivative gain

k_i - Inertia of link i elements about their centers of mass

k_τ - motor current-torque constant

l - surface following projection

$\mathbf{L}(s)$ - control loop transfer function

$\mathbf{L}_t(s)$ - target control loop transfer function

$\mathbf{M}(\mathbf{z})$ - mass matrix

$\tilde{\boldsymbol{M}}$ - linearized mass matrix

$\mathbf{M}_f(s)$ - recovery matrix

m_{hi} - mass of hub i

m_{ti} - mass of tip i

m_{li} - element masses of link i

m_{Li} - total mass of link i

Mx,y,z - Moment about the x, y, or z axes

$\mathbf{n_x}$ - surface orientation tangential vector

$\mathbf{n_y}$ - surface orientation normal vector

o_{hi} - distance from shaft i to link i attachment to hub

o_{ti} - distance from tip attachment point of link i to shaft i+1

q_{ij} - jth element rotational deflection

$\mathbf{P}(s)$ - plant transfer function

p_x - x-position

p_{xci} - ith commanded x-position

p_y - y-position

p_{yci} - ith commanded y-position

$\mathbf{Q}$ - state penalty matrix

r - radius

$\mathbf{R}$ - torsional damping matrix; control penalty matrix

s - second (unit of time)

$\mathbf{s}$ - complex frequency

$\mathbf{S}$ - solution to an Algebraic Riccati Equation

t - time

$\mathbf{T}$ - tangential direction vector

Tslew - slew time

$\mathbf{u}$ - generalized speed vector; control input

u_{fx} - x-force control input

u_{fy} - y-force control input

$\mathbf{v}(t)$ - sensor noise vector

$\mathbf{V}(\mathbf{z},\mathbf{u})$ - Coriolis and centripetal matrix

$\mathbf{w}(t)$ - process noise vector

$\tilde{X}_\infty$ - optimal state error covariance

x_c - arc x center location

x_{ep} - global x Cartesian tip position

y_c - criterion output; arc y center location

y_{ep} - global y Cartesian tip position

y_s - sensor output

$\mathbf{z}$ - manipulator position vector

Γ - noise input coupling matrix

τ_i - motor torque at joint i

θ_i - base rotation of link i

“ - inches

Acknowledgments

I would first and foremost like to thank my primary advisor, Professor Juris Vagners, for all of his efforts to help me complete my dissertation in a timely manner. If I were to sit down with a pen and paper and try to design the “perfect advisor” for me, there is no way I could have come up with a better one than Juris Vagners! His ability to wade through the unnecessary material and cut to the chase of what is important is unparalleled. Also, his sense of humor and genuine concern for each of his students are impressive.

Of course, I’d also like to extend a great deal of thanks to Professor Uy-Loi Ly for his modern controls and SANDY expertise. On many occasions I’d come to him stumped and he ALWAYS had an answer. There is no doubt that a large part of the success accomplished in this dissertation is due to Professor Ly.

Thanks also goes to Professor Martin Berg for his support of the actual two-link arm hardware. In addition, he served as a member of my supervisory committee. Professors Eli Livne and Deirdre Meldrum also deserve thanks for serving as members of my supervisory committee as well.

A special thanks goes to The Boeing Company for providing the force sensor used in this research.

I’d also like to thank all the friends I’ve made here over the last 3 years. They’ve helped make this a very enjoyable experience. I’d like to single out Kalev Sepp, Vatchara Lertpiriyasuwat (yes, that’s actually his name!), and Dave Rathbun for their help in the lab. A special thank you also goes out to Steve Evers for actually putting the arm hardware together. His excellent work made my research go very smoothly.

Finally, I’d like to thank my wife, Nina, for all of her support during these 3 years. I’d be remiss if I didn’t thank my daughter Katrina and my son Brett as well.

I’d like to end my acknowledgments with a quote from Isaiah (26:12):

“You grant us prosperity Lord; Everything that we achieve is a result of what you do.”

1 Introduction

1.1 Motivation

An area of importance in manufacturing applications is hybrid position and force control. For tasks such as buffing, polishing, grinding, painting, etc. the force applied by the tool to the workpiece needs to be controlled while simultaneously controlling the position. This hybrid control is also known as constrained motion since the manipulator is constrained by a surface.

In addition, the tool in a manufacturing task must start at an arbitrary location in free-space, move to the workpiece, perform the task, and then return to the starting location. This complete or unified task must thus include free-space motion to get to the workpiece, impact control to successfully make contact with the object, constrained hybrid position and force control to perform the task, and some way to transition back to free space motion.

Many of the current applications for manipulators consider only the rigid body dynamics of the robot. However, all rigid bodies have some vibrational or flexible modes associated with them. This is especially true for lighter weight manipulators which have lower frequency modes of vibration, especially in a high-speed application. This leads to an issue of flexibility, and controlling the flexible modes of lightweight manipulators is an area of keen interest. Thus, a good design must address flexibilities in the manipulator. In the future, applications will require more of the robotic manipulators as performance requirements increase.

Another common practice in control of industrial manipulators is the use of full-order controllers. Full-order controllers have the disadvantage of being the same order as the model and can thus take up large amounts of memory and involve excessive computational time. Reduced-order controllers designed through direct-optimization allow use of the full-order model without the penalty of increased controller size.

The approach taken here is to use the SANDY direct-optimization design tool to simultaneously design robust, reduced-order controllers. Also, this optimization allows the need for position and force selection matrices to be considered in the optimization of the controllers through bandwidth separation

Finally, most manufacturing tasks are completed with the complete knowledge of the workpiece shape. Sometimes, it takes a great deal of time for the exact shape to be programmed. In addition, each piece usually varies somewhat and requires reworking of the feedforward position based on the shape. An alternative to this is to use an algorithm which follows an arbitrary shape with no a priori knowledge. Thus, the same controllers and feedforward references could be used for many surfaces.

This research seeks to consider all the issues discussed above. Thus, a complete (unified) hybrid control task using robust reduced-order controllers to control a flexible two-link manipulator while working on an unknown surface at an unknown location is analyzed, developed, and implemented. The test bed for the research is the University of Washington Control Systems Laboratory's flexible two-link manipulator, or two-link arm (TLA) for short.

1.2 Contributions

The contributions of this dissertation include the following:

1. Interfacing the TLA manipulator to the rapid-prototyping AC100-C30 real-time data acquisition and control system.

2. Performing parameter identification of the TLA manipulator, and performing experimental validation of the resulting model for both position and force control.

3. Developing a baseline of performance for the TLA manipulator using Proportional-Integral-Derivative (PID) position and force control.

4. Designing, simulating, and experimentally comparing full-order Linear-Quadratic Gaussian (LQG), LQG/Loop Transfer Recovery (LTR), and Special Coordinate Basis LTR position controllers.

5. Designing, simulating, and experimentally comparing reduced-order LQG, LQG/LTR and SCB/LTR controllers to full-order position controllers.

6. Designing, simulating, and experimentally comparing full-order and reduced-order LQG force controllers.

7. Designing, simulating, and experimentally evaluating direct non-linear opti-

mization SANDY position and force controllers.

8. Designing, simulating, and experimentally validating robust reduced-order position and force controllers for the TLA.

9. Developing an impact control algorithm to allow successful touchdown of the TLA onto a surface whose exact location is not known.

10. Developing a complete (unified) control structure to allow free-space motion, surface touchdown, constrained motion, transition to free-space motion, and return to the initial starting location.

11. Comparing hybrid direct-optimization designs by superposition and simultaneous optimization.

12. Developing a surface following algorithm to allow the TLA to follow a surface without a priori knowledge of the surface while maintaining a constant force.

1.3 Organization

This dissertation is divided into 16 chapters and 5 appendices. Before starting into the new work associated with this dissertation, a literature review of current flexible manipulator force and position control is presented in Chapter 2. After establishing the current research, the experimental setup for this research is described in Chapter 3. Next, the dynamics model for position control is discussed in Chapter 4, leading to the PID control design and experiments presented in Chapter 5. Continuing, the position design is extended to LQG-class controller designs and experiments in Chapter 6. Another position control design technique, SANDY direct-optimization design, is described in Chapter 7 and the subsequent position control application is presented in Chapter 8.

The position control design is extended to an end-point framework with SANDY in Chapter 9 before the dissertation covers a hybrid control overview in Chapter 10. Next, Chapter 11 explains the force control model and design while Chapter 12 discusses the actual hybrid control designs. Logically following the hybrid control design is the surface following experiment coverage in Chapter 13. After successfully following a surface while starting in contact, the process of surface touchdown is explained in Chapter 14. Finally, the complete hybrid task is described in Chapter 15 while conclusions are presented in Chapter 16.

The appendices contain additional information on several topics that might be of interest to some readers. Appendix A contains the arm modelling details not included in Chapter 4 while Appendix B presents additional position experiment plots for the SANDY and PID controllers. Similarly, additional LQG-class position experiment plots are found in Appendix C while Appendix D displays the additional end-point tracking plots from Chapter 9. Finally, Appendix E presents some of the MATLAB m-files developed for this work.

2 Literature Review

In order to give credit to appropriate people, and to avoid duplication of previous work, an extensive literature search is done. In particular, a review of all work on multiple-link flexible robotic manipulator theory, simulation, and experimentation is accomplished. In addition, force control, hybrid force and position control, and surface following applications are examined.

Flexible robotic manipulators have been studied a great deal recently due to more stringent requirements. Space structure applications usually require low-weight due to the high cost of sending heavy payloads into space. Space manipulators must be light for the same reason, and therefore flexibility issues become important. One example of an existing flexible space manipulator is the Shuttle Remote Manipulator System built by Spar Aerospace Limited for the Space Shuttle. Some of the new requirements driving flexible manipulator technology are lower arm cost, higher motion speeds, and improved mobility [4].

2.1 Current Flexible Manipulator Facilities

Rigid manipulator control has been well-studied by many people [6] [40] [55]. The key difference between the traditional rigid manipulators and the lightweight flexible manipulators lies in the elastic modes associated with the flexible manipulators. For this reason, many flexible manipulator facilities have been established. Much of the current research has been done at Stanford University [56] [57], and the University of Washington's Two-Link Arm (TLA) is based on Stanford's design. Among the other locations with experimental planar two-link flexible manipulators are Martin Marietta [65], Sandia National Laboratories [21], the USAF Phillips Laboratory [19], and the Polytechnic Institute [33].

2.2 Current Flexible Manipulator Applications

There have been many contributors to flexible manipulator research. Some approaches have focused on adaptive control [4] [24] [13]. Others have focused on linear programming [66], self-tuning controllers [32], sliding-mode control [14], nonlinear model-based control [58], and gain-scheduled H_∞ controllers [51]. In addition, use of rigid-body based controllers with input preshaping for flexible manipulators has been investigated [33], as well as control of flexible arms with friction in the joints [23]. Also, some studies have focused on inversion-based controllers [43]. Stanford University has also been very active in the area of flexible manipulator end-point control [57].

2.3 Constrained Motion Control

A robot working in contact with a surface is called a constrained motion problem if the surface is rigid, or a "compliant motion task" for a compliant surface [53]. There has been a great deal of work on the constrained motion problem. In fact, a literature search on "force control" returned over 1500 documents! The two major areas are impedance control [16][37] and hybrid position and force control [61]. The first force control stability analysis, found in [67], showed that a trade-off exists between the force gains and the environment stiffness. Others attribute the dynamic instability of force control to higher-order dynamics and conclude that there must be some compliancy in either the robot or the environment [1]. A stable hybrid approach based on the original hybrid scheme was developed by Fisher [25]. This "correct formulation" uses pseudo-inverse matrices to calculate the angle error vectors.

Another approach developed by Matsuno [49] applied perturbation techniques to a multi-link flexible manipulator. Also, Cai [12] used a Lyapunov approach on a rigid manipulator to design his hybrid control scheme. In addition, Kwan used neural nets to tackle the constrained motion problem [36].

Stanford has been very involved in force control research for flexible manipulators. Maples [48] conducted force control research on a single-link flexible manipulator. This was extended by Kraft [34], who added a mini-manipulator on the end of a single-link manipulator and experimentally implemented force and position control algorithms. The most recent work by Ballhaus [3] extended the force and position control to a two-link flexible manipulator with a mini-manipulator on the end. All of this research was done using full-order LQG regulators, and no effort was made to minimize the order of the controllers. Another method uses a transformation to simultaneously control both position and force [29].

Also, there has been some work with the surface touchdown problem, sometimes called impact control. One approach decreased the initial force command by input command preshaping [38]. Another implemented a discontinuous controller to reduce the problems of impact stability [53]. Finally, another paper considered a compliant wrist and tunable PID controller [47].

Kraft's approach [34] involved independently designing position and force controllers, and then just adding the torque commands together to produce one hybrid signal. Kraft's approach is similar to basic hybrid control described in [61], but uses a low-friction roller to measure the surface normal (and surface tangential direction) directly instead of using selection matrices. This is done because it is easier to track surfaces without prior knowledge of the surface shape, which was the object of his research.

2.4 Surface Following Applications

There are many applications where a robotic manipulator is required to follow a surface while applying a constant force to perform a task. These applications include buffing, grinding, polishing, or painting. The normal approach to tackle these problems is to learn the desired trajectory and then perform the task. The obvious problem here is that every new shape requires new trajectory learning.

Some of the current methods for learning the trajectory are learning control [42] and genetic techniques [27]. Another approach for surface uncertainty is fuzzy compliance control [60]. Others have focused on trying to follow a surface through differential workpiece models [18]. Finally, some have tried to follow an unknown surface using range sensors [59] [54].

A promising approach involving hybrid force and position control and was developed by Kraft [34] for a single-link flexible manipulator with a mini-manipulator mounted at the end. This approach uses a low-friction roller which senses essentially the surface normal force and then commands a position reference along the surface tangent.

2.5 The Complete (Unified) Hybrid Task

The complete or unified hybrid task involves starting at a point in free-space, acquiring the workpiece during the impact stage, performing the hybrid or constrained task, and then returning to free-space motion. Several researchers have considered this area. Mandal and Payandah [47] conduct an extensive review of various components of the unified task, and tie the stages together with a knowledge-based tuning of gains. Also, Khatib and Burdick [31] propose a unified approach based on an operational space formulation. These and others are either complex or require different controllers for each stage.

2.6 Summary

The preceding sections show that flexible manipulator dynamics and position control have been widely studied. Also, many position control techniques have been applied to flexible multi-link and rigid manipulators. None of the work surveyed considered robust reduced-order position control of flexible two-link manipulators.

Force control, also, has been widely studied. Force control, when used in conjunction with position control at a surface is known as the constrained motion problem. Many techniques have been studied, but once again none have considered robust reduced-order designs.

Finally, the complete or unified hybrid tasks surveyed required modification of the controllers for each phase of the task or were very difficult to implement. Thus, a simple unified approach which uses the same controllers for all the phases would be beneficial.

3 Experimental Setup

3.1 General Description

The University of Washington Control Systems Laboratory Two-Link Arm (TLA) Manipulator consists of a shoulder motor, an elbow motor, elbow and wrist plastic air support pucks, an end effector assembly, and two links connecting the motors and end-effector. The configuration is mounted to a 4' by 8' granite table. Any combination of rigid and flexible links is possible. In addition, it is possible to run the system as a single link. Also, to provide a low friction environment, both ends of the forearm are supported by pressured air coming out of plastic pucks at the elbow and wrist [22]. Figure 3.1 shows a block diagram of the TLA hardware setup while Figure 3.2 displays a picture of the TLA facility.

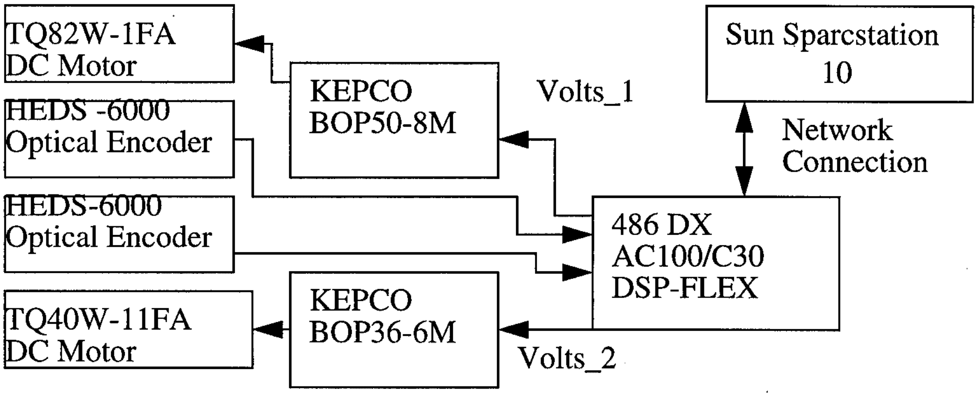


Figure 3.1- Block Diagram of TLA Hardware

3.2 Actuators

Both motors are brushless DC torque motors with effectively no backlash. The shoulder motor is an Aeroflex TQ82W-1FA and the elbow motor is an Aeroflex TQ40W-11FA. Each DC motor consists of a stator and a permanent magnet. Each stator consists of magnetic conduction cores wrapped by current carrying conductors in a toroidal fashion. The magnet acts as the rotor, and is separated from the stator by a small air gap.

The brushless DC motors are simple 2 pole devices. The current running through

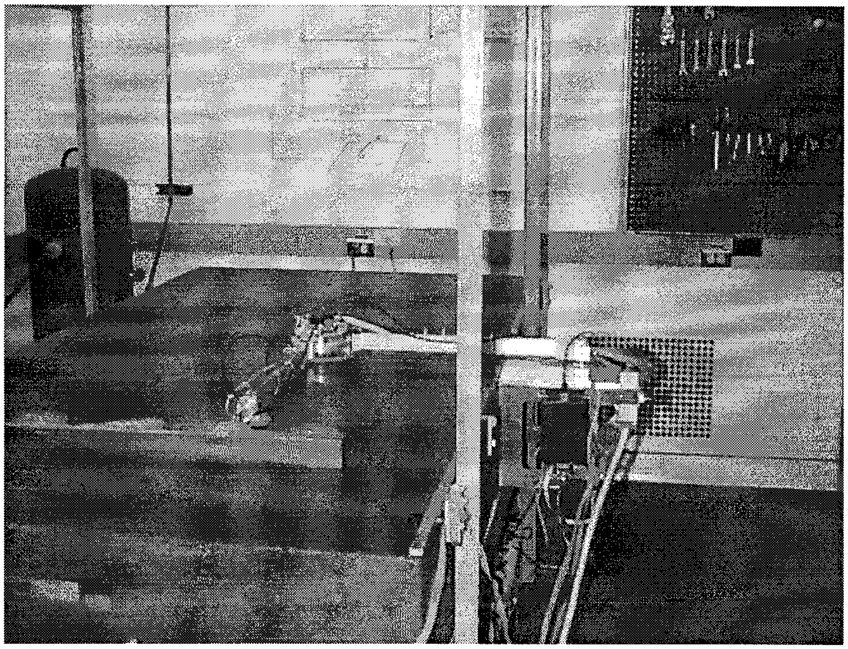


Figure 3.2- TLA Facility Picture

the stator sets up a magnetic field and the permanent magnet tries to align itself with the field under a developed actuator torque $\tau(t) = K_{\tau}(\theta) i(t) \varepsilon$. The motors require sourcing of the winding current in either a voltage or current control mode. For ease of interface to and isolation from the control signal, current control mode is used.

The power amplifiers which drive the motors are the KEPCO BOP50-8M for the shoulder motor and the KEPCO BOP36-6M for the elbow motor. They are linear amplifiers which can operate in either voltage or current mode and can sink 100% of their current rating at any voltage setting.

Since the models used in this research are in terms of Newton-meters and the actual output from the data acquisition system to the actuators is in volts, a conversion factor is required. Figure 3.3 shows the conversion factors for actuator 1 and 2 (the shoulder and elbow motors, respectively).

3.3 Links

Four basic link configurations exist: SLA_A, SLA_B, TLA_A, and TLA_B.

SLA_A is a single flexible link with the tip bracket centered on the tip pad, while SLA_B is also a single flexible link, but with no tip load. Configuration TLA_A consists of two flexible links, with the elbow stator tied to the first link, and TLA_B also two flexible links, but with the elbow stator tied to the second link. In addition, a set of very flexible links has been manufactured for use with the two link configuration, but is not used in this research.

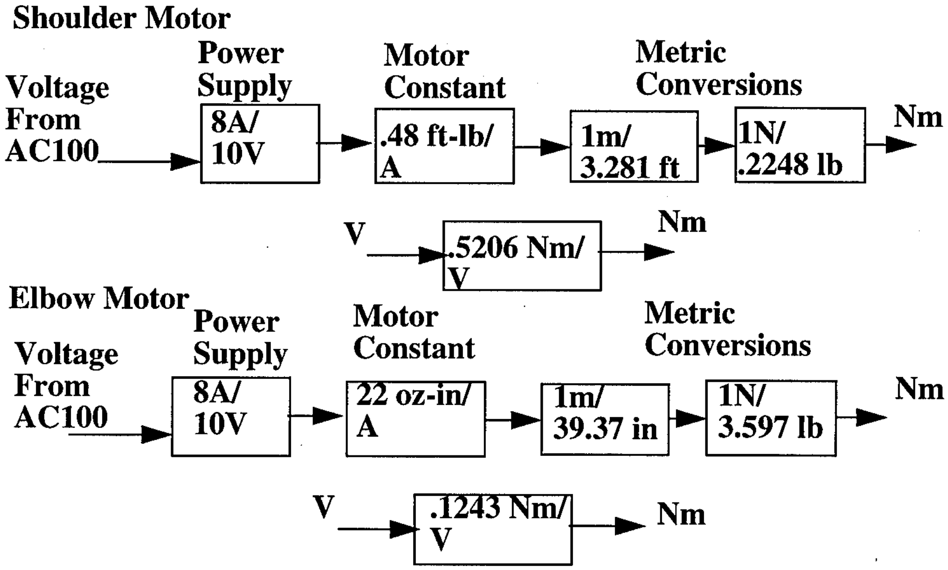


Figure 3.3- Actuator Conversion Factors

3.4 Sensors

The joint angle signals are produced by optical encoders mounted on the DC motor shafts. Since the DC motors are direct drive, a high degree of accuracy is available from these collocated sensors. Each joint is fitted with a Hewlett-Packard HEDS-6000 optical encoder. The encoder provides 3 channels of output, namely two channels for quadrature square waves and one channel for an index pulse. Both channels A and B provide 1024 cycles per revolution. The quadrature lines and index line are fed directly to a quadrature decoding module in the AC100 data acquisition system.

In addition, a proximity sensing system is used to enable the controller to know when the surface is close. Such knowledge enables the controller to slow the arm down and prevent excessive speed and force upon touchdown. A Microswitch MPF5 analog photoelectric control is used as the proximity sensor. This device is used in conjunction with an MPS35 sensor base, an MPB11 receptacle, and an FE-B5B-5 fiber optic cable to provide an analog output proportional to the straight-line distance from a surface. This is a fiber optic system which transmits a beam of infrared light out of one part of the cable and receives the reflected light back in the other part of the cable. The amount of reflected light received back is directly proportional to the distance between the cable end and a surface. The active range of this sensor is 1-4 inches, so by placing the cable end 1 inch behind the roller attached to the TLA wrist, the readout is effectively a voltage from 0-5 Volts which is proportional to 0-3 inches. This particular sensor may be used in further research on the TLA.

Also, a 6-component force sensor mounted on the wrist provides force readings in the x, y, and z directions, as well as moments about the x, y, and z axis. The Fx and Fy readings are used in the force control algorithm. Specifically, the sensor system is an Assurance Technologies Incorporated Mini 5/10 system, shown in Figure 3.4.

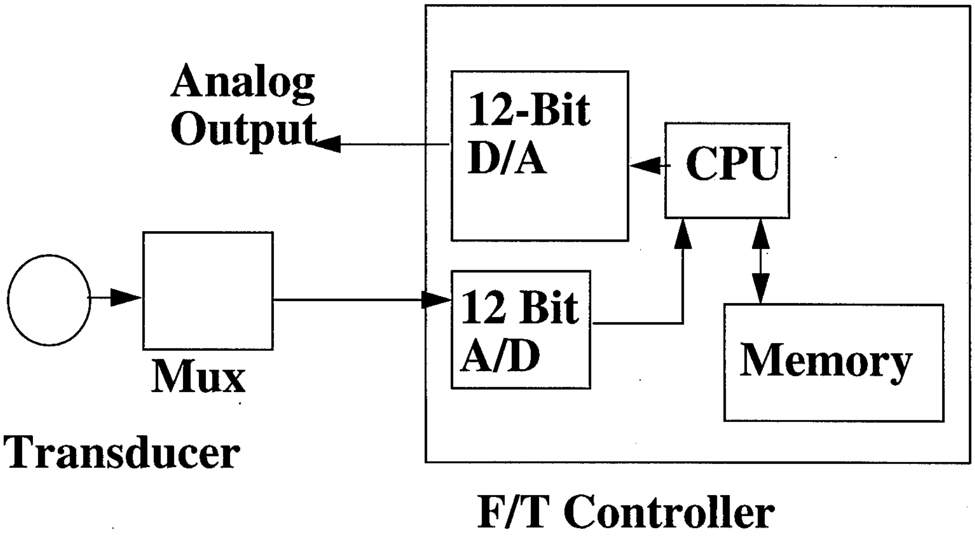


Figure 3.4- ATI Force Sensor Block Diagram

This system includes a 5 lb/10 in-lb force/torque sensor, and multiplexor and signal conditioning unit, and a controller. The controller reads in the raw strain gauge voltages, converts them to a digital signal, applies a 6x6 calibration matrix, and produces analog outputs directly proportional to the actual forces and torques. The sensor is attached to the end of the second link on one side and to a low-friction roller on the other side, shown in Figure 3.5. This enables the sensor to detect primarily only the surface normal. The overall sensor has the capability shown in Figure 3.6.

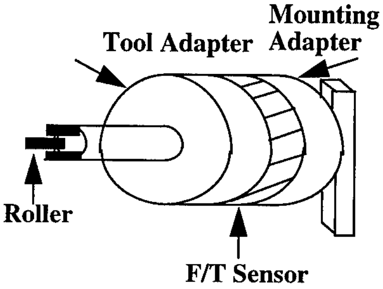


Figure 3.5- Force Sensing Mechanism

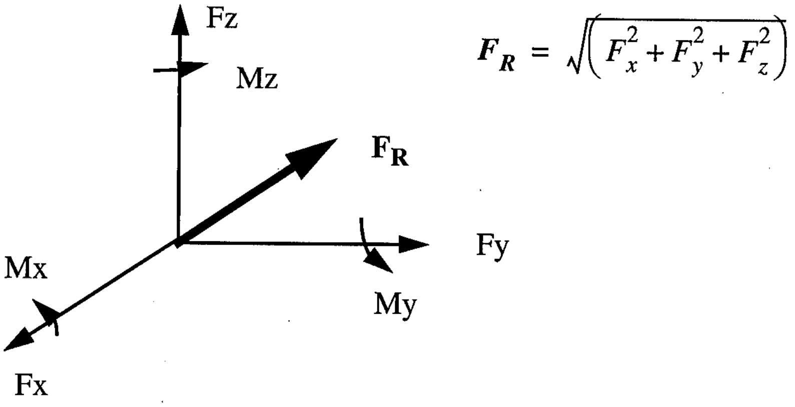


Figure 3.6- Force/Torque Conventions

3.5 Data Acquisition System

The data acquisition system used for the experiments is an ISI AC100/C30 system. This system operates at 100 Khz and interfaces with MATRIXx System Build directly through an autocode generator. The autocode generator translates System Build block diagrams into C code which interfaces the hardware to the arm. The computational platform is a Sun Sparcstation/10.

The AC100/C30 possesses IP-DIG24, IP-DAC, IP-QUAD, and IP-HiADC cards which are all installed on a DSPFLEX board located in a 66 MHz 486 PC. The IP-DIG-24 provides 24 digital (TTL) input/output channels, while the IP-DAC provides 6 12-bit analog output channels configured to +/-5V or -10V. In addition, the IP-QUAD module provides 4 quadrature decoding channels which can detect either position or rate from quadrature signals. Finally, the IP-HiADC possesses 12 input analog channels with 12-bit resolution and synchronous sampling of all inputs.

3.6 Software Interface

MATRIXx System Build is the software interface for the system. This system allows interface between the controller block, the interaction block, and the plant block. Each of these blocks is a superblock, which contains individual functions displayed as boxes inside. In this way, the designer gets a clear picture of the interface between the individual components involved in the design. Figure 3.7 shows the block diagram for the overall simulation environment.

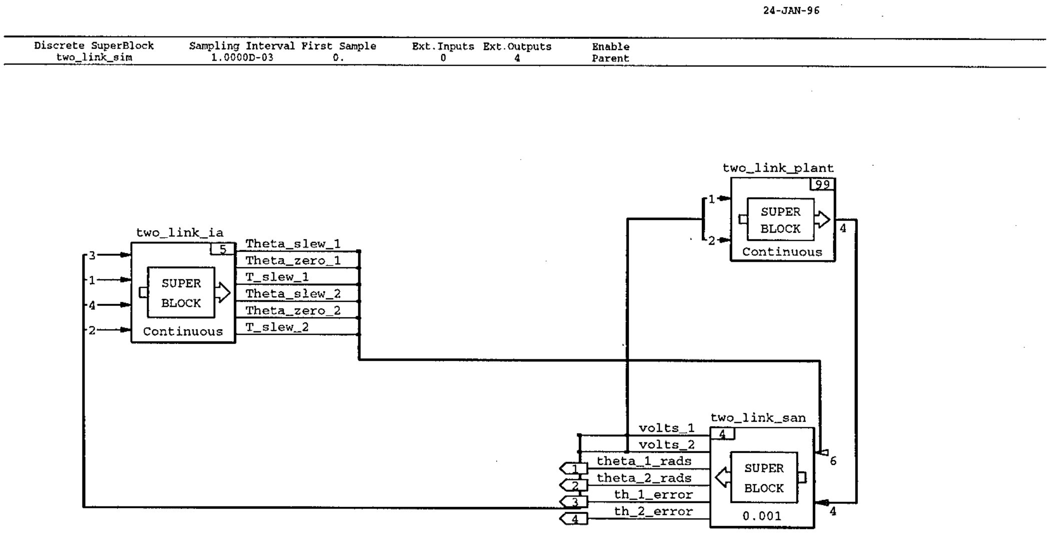


Figure 3.7- System Build Simulation Block

The interactive animation block allows the user to change the reference trajectory path “on-the-fly”. The user may specify the start angle, end angle, and slew time for the reference trajectory while viewing the angle and voltage (torque) outputs real-time. shows a block diagram of the interactive animation block. Figure 3.8 shows the interactive animation block diagram.

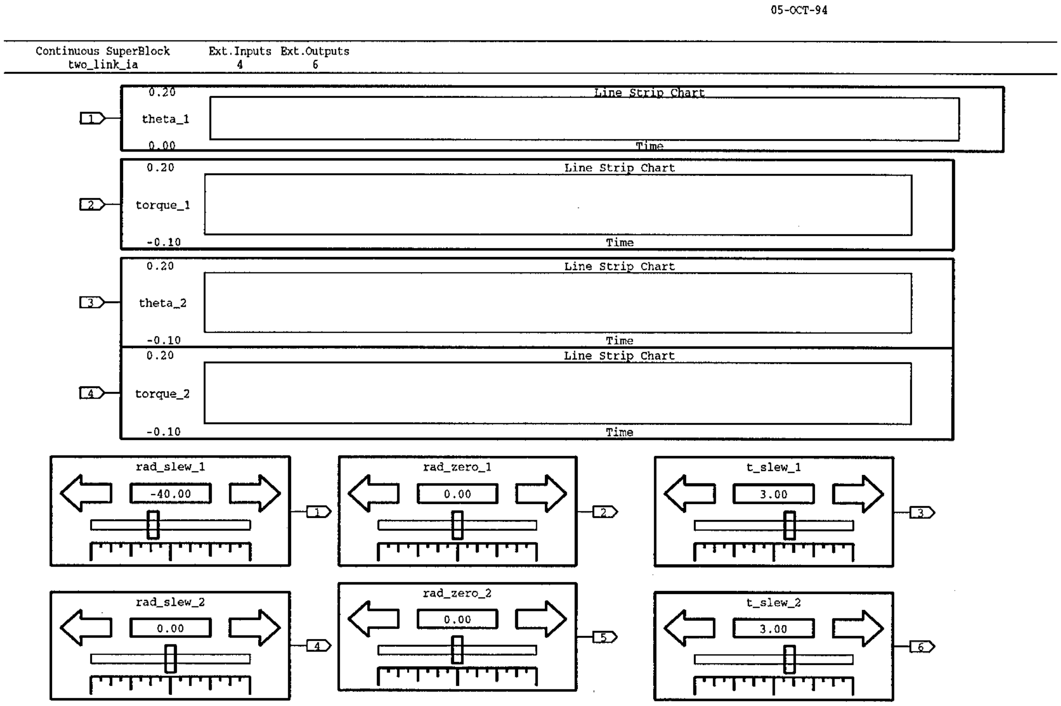


Figure 3.8- Interactive Animation Block Diagram

The plant simulation block is implemented with a state space model. In addition, the Newton-meter to voltage inverse conversion is needed to account for this scaling in the controller block. Finally, the angle output must be converted from radians to counts to match the optical encoder and quadrature decoder sensor. The block diagram is shown in Figure 3.9.

Lastly, the typical controller is implemented by the feedback structure shown in Figure 3.10. Here, the slew command is compared to the actual angle and the error is fed into the controller. For this setup (the SANDY controller, to be discussed later), the controller has two inputs (the angle errors) and two outputs (the torques). These torques are converted from Nm to Volts (the hardware output units).

Appendix E contains listings of the Matlab m-files used to aid in completion of this dissertation. In particular, the model generation, SANDY and LQG design, and bandwidth m-files are included.

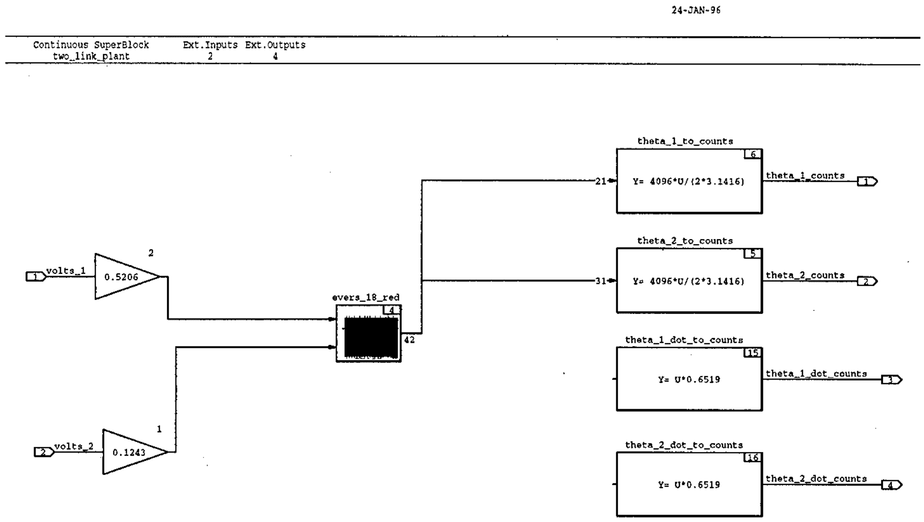


Figure 3.9- Simulation Plant Implementation

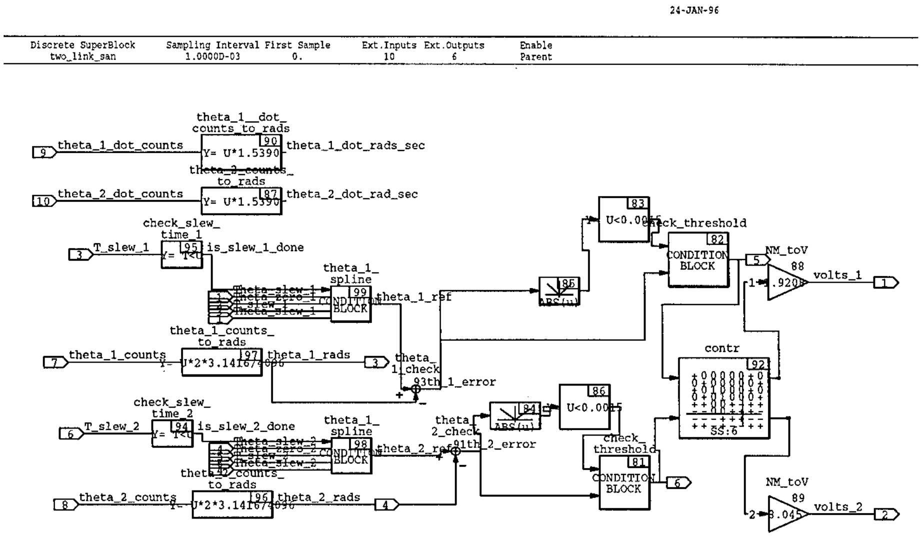


Figure 3.10- Typical System Build Controller Structure

4 Position Dynamics Model

The model used in this research is a lumped-spring-mass model based on the work of Oakley [56]. Appendix A provides the details of the model, including a complete 6 degree of freedom (DOF) example of the complete non-linear equations of motion for a two-link manipulator. In addition, Appendix A also gives the parameter descriptions and their values. For a two-link arm, 6 DOF corresponds to 3 elements for each link. The model used for this research assumed a 20 DOF model, or 10 elements per link. This corresponds to 20 velocity states and 20 position states, or a 40th order model. The model also includes 2 additional outputs for x and y end-point positions through the use of forward kinematic equations.

4.1 Non-Linear Equations of Motion

The nonlinear equations of motion are developed by Oakley [56] based upon Kane's method [30]. It is convenient to express a generalized speed vector **u** as:

$$\boldsymbol{u}_{i_1} = \dot{\theta}_1 + \sum_{k=1}^{i_1-1} \dot{q}_{1k} ; i_1 = 1, \ldots, n_1 \tag{4.1}$$

$$\boldsymbol{u}_{n_1+i_2} = \dot{\theta}_1 + \sum_{k=1}^{n_1-1} \dot{q}_{1k} + \dot{\theta}_2 + \sum_{k=1}^{i_2-1} \dot{q}_{2k} ; i_2 = 1, \ldots, n_2 \tag{4.2}$$

$$\boldsymbol{u} = [u_1, \ldots, u_{n_1}, u_{n_1+1}, \ldots, u_{n_1+n_2}]^T \tag{4.3}$$

where $\dot{\theta}_1$ is the shoulder motor velocity, $\dot{\theta}_2$ is the elbow motor velocity, $\dot{q}_{1k}$ is a sublink velocity for link 1 and $\dot{q}_{2k}$ is a sublink velocity for link 2. Additionally, the **z** vector contains the position of the manipulator and the τ vector contains the motor torque.

$$z = [\theta_1, q_{11,\ldots}, q_{1n_1}, \theta_2, q_{21}, \ldots, q_{2n_2}]^T \tag{4.4}$$

$$\tau = [\tau_1, \tau_2]^T \tag{4.5}$$

The complete non-linear matrix equation for a two-link manipulator is therefore represented by the following equation:

$$\boldsymbol{M}(z)\,\dot{\boldsymbol{u}} + \boldsymbol{V}(z, \boldsymbol{u}) + \boldsymbol{R}\dot{z} + \boldsymbol{K}z = \boldsymbol{T}\tau \tag{4.6}$$

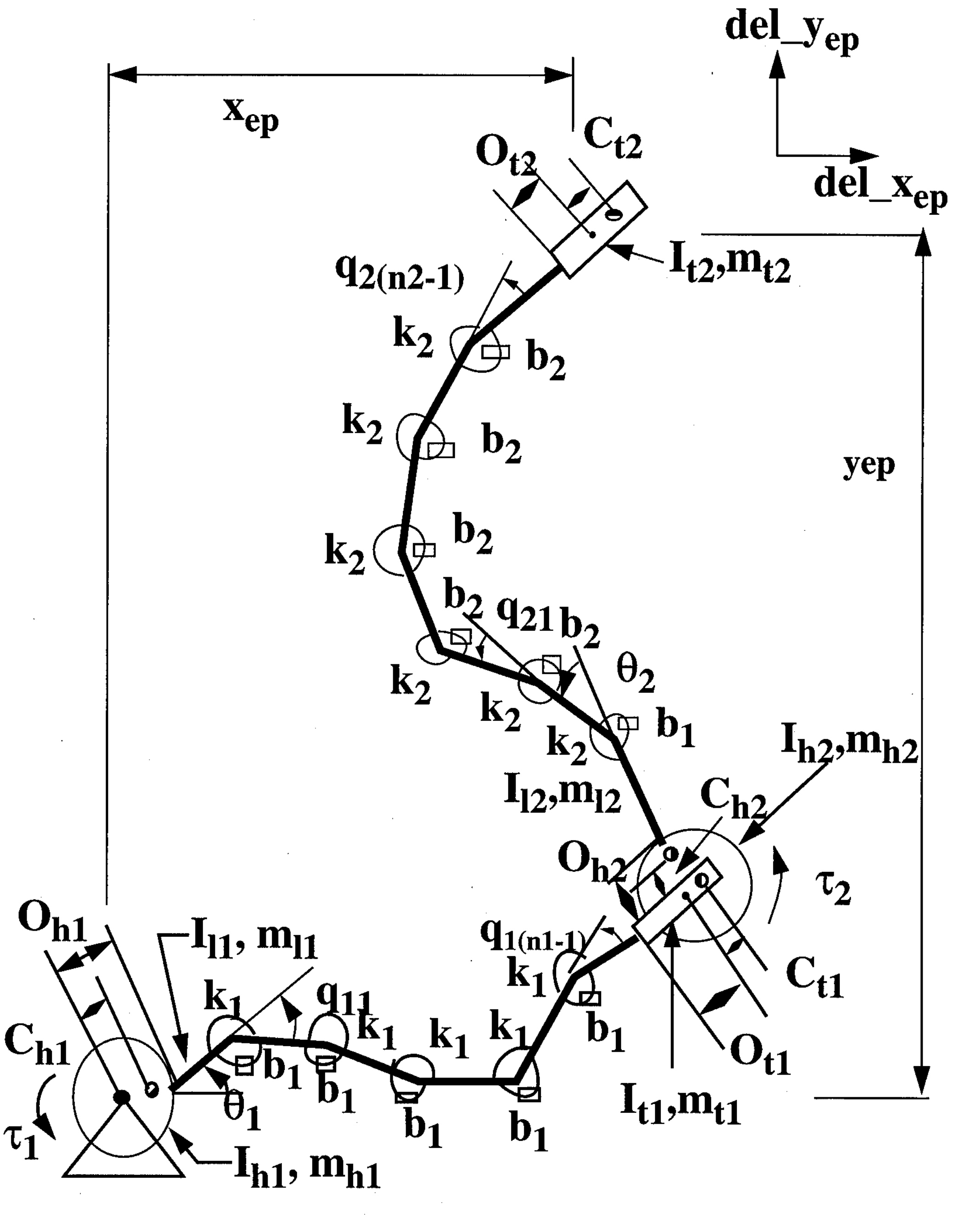


Figure 4.1- Lumped-Spring-Mass-Damper Model Parameters

where **M** possesses the mass and inertia properties, **V** contains the centripetal and Coriolis terms, **R** represents the Coulomb torsional damping, and **K** contains the torsional stiffness. Also, **T** is the torque input distribution matrix.

4.2 Linearized Equations of Motion

Linearization may occur about any joint condition with zero link deflections and all velocity states equal to zero. Application of a Taylor's series expansion about the non-linear equations of motion in equation (4.6) produces the desired linearization. The resulting linearized equation of motion is:

$$\tilde{\boldsymbol{M}}\Delta\ddot{\underset{\sim}{z}} + \boldsymbol{R}\Delta\dot{\underset{\sim}{z}} + \boldsymbol{K}\Delta\underset{\sim}{z} = \boldsymbol{T}\Delta\underset{\sim}{\tau} \qquad (4.7)$$

where $\tilde{\boldsymbol{M}}$ is a Taylor expansion of **M** about the chosen linearization condition. **V** from the non-linear equation disappears since it is a function of velocity squared, and velocity is assumed small near the linearization state.

It is convenient to place the linearized equations of motion in the state space form of the following equations.

$$\begin{bmatrix} \tilde{\boldsymbol{M}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} \Delta\dot{\underset{\sim}{z}} \\ \Delta\underset{\sim}{z} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{R} & -\boldsymbol{K} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\dot{\underset{\sim}{z}} \\ \Delta\underset{\sim}{z} \end{bmatrix} + \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{0} \end{bmatrix} \Delta\underset{\sim}{\tau} \qquad (4.8)$$

$$\frac{d}{dt} \begin{bmatrix} \Delta\dot{\underset{\sim}{z}} \\ \Delta\underset{\sim}{z} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{M}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} -\boldsymbol{R} & -\boldsymbol{K} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\dot{\underset{\sim}{z}} \\ \Delta\underset{\sim}{z} \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{M}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{0} \end{bmatrix} \Delta\tau \qquad (4.9)$$

$$\dot{\underset{\sim}{x}} = \boldsymbol{A}\underset{\sim}{x} + \boldsymbol{B}\Delta\underset{\sim}{\tau} \qquad (4.10)$$

$$\underset{\sim}{y} = \boldsymbol{C}\underset{\sim}{x} + \boldsymbol{D}\Delta\underset{\sim}{\tau} \qquad (4.11)$$

$$\boldsymbol{A} = \begin{bmatrix} \tilde{\boldsymbol{M}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} -\boldsymbol{R} & -\boldsymbol{K} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \qquad (4.12)$$

$$\boldsymbol{B} = \begin{bmatrix} \tilde{\boldsymbol{M}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{0} \end{bmatrix} \qquad (4.13)$$

$$\underset{\sim}{x}^{T} = \begin{bmatrix} \Delta\dot{\underset{\sim}{z}}^{T} & \Delta\underset{\sim}{z}^{T} \end{bmatrix}^{T} \qquad (4.14)$$

Matlab m-files tlapar.m and tlagen.m achieve the linearization using this scheme. Also, the m-files are converted to MATRIXx form in tlapar.mws and tlagen.mws to achieve the linearization. The tlapar.m file contains the parameters which are specific to the manipulator model, such as the hub inertia, as well as containing the number of elements per link desired. After loading the parameters, running the tlagen.m file actually generates the new A and B matrices, and calls them Abar and Bbar. The input to the model is the torque vector and the output contains the states as well as x and y endpoint positions. The original work for the m-files was done by Evers [22].

4.3 End-point Calculations

The end-point calculations are not explicitly part of the equations of motion. However, the x and y end-point positions are provided as the last 2 outputs from the model for use in simulation and control. The equations are:

$$\begin{aligned}
x_{ep} = {} & o_{h1}\cos(\theta_1) + \sum_{i_1=1}^{n_1-1} l_1 \cos\left(\theta_1 + \sum_{k=1}^{i_1} q_{1k}\right) + \ldots \\
& \ldots + o_{t1}\cos\left(\theta_1 + \sum_{k=1}^{n_1-1} q_{1k}\right) + o_{h2}\cos\left(\theta_1 + \sum_{k=1}^{n_1-1} q_{1k} + \theta_2\right) + \ldots \\
& \ldots + \sum_{i_2=1}^{n_2-1} l_2 \cos\left(\theta_1 + \sum_{k=1}^{n_1-1} q_{1k} + \theta_2 + \sum_{k=1}^{i_2} q_{2k}\right) + \ldots \\
& \ldots + o_{t2}\cos\left(\theta_1 + \sum_{k=1}^{n_1-1} q_{1k} + \theta_2 + \sum_{k=1}^{n_2-1} q_{2k}\right)
\end{aligned}$$

$$\begin{aligned}
y_{ep} = {} & o_{h1}\sin\theta_1 + \sum_{i=1}^{n_1-1} l_1 \sin\left(\theta_1 + \sum_{k=1}^{i_1} q_{1k}\right) + \ldots \\
& \ldots + o_{t1}\sin\left(\theta_1 + \sum_{k=1}^{n_1-1} q_{1k}\right) + o_{h2}\sin\left(\theta_1 + \sum_{k=1}^{n_1-1} q_{1k} + \theta_2\right) + \ldots \\
& \ldots + \sum_{i_2=1}^{n_2-1} l_2 \sin\left(\theta_1 + \sum_{k=1}^{n_1-1} q_{1k} + \theta_2 + \sum_{k=1}^{i_2} q_{2k}\right) + \ldots \\
& \ldots + o_{t2}\sin\left(\theta_1 + \sum_{k=1}^{n_1-1} q_{1k} + \theta_2 + \sum_{k=1}^{n_2-1} q_{2k}\right)
\end{aligned} \tag{4.15}$$

4.4 Low-Torque Model

Model parameter identification was achieved through applying 3 different slew trajectories to 2 different Proportional-Integral-Derivative (PID) controllers, and then choosing the damping and flexural rigidity terms to best match the experimental response. Table 4.1 shows the PID controller gains used to perform parameter identification for the low-torque model. When the applied torque onset is relatively slow, friction effects tend to increase the damping and lower the natural frequency.

Table 4.1 - PID Parameter ID gains

Controller	P	I	D
1	.5	.01	.01
2	.5	.01	.01

Different controllers and trajectories were used to ensure the model was not tuned to one controller or trajectory. A relatively high PID gain was used for controller 1 to provide many overshoots for ease of damping identification, while a lower gain was used for controller 2 to provide insight into bearing stiction effects at lower gains. The 3 test slews were chosen to excite dynamics for moving each joint individually while holding the other constant as well as checking the dynamics for slewing both links at the same time. Table 4.2 displays the 3 test slews used for parameter ID and Table 4.3 contains the resulting values of these tuned parameters.

Table 4.2 - Test Slews for Parameter ID

Slew	θ1_start (deg)	θ1_end (deg)	θ2_start (deg)	θ2_end (deg)
1	40	0	90	90
2	0	0	110	70
3	20	0	110	90

Table 4.3 - Experimentally Tuned Values for the Low-Torque Model

Link	b	EI
1	8	.225
2	.1	.3

These tuned values provide a model with eigenvalues listed in Table 4.4.

Table 4.4 - Low-Torque Model Eigenvalues

High Torque Eig	High Torque Damping	High Torque Freq (rad/s)
0	1	0
0	1	0
0	1	0
0	1	0
5.1174e-01	1	5.1174e-01
5.1174e-01	1	5.1174e-01
5.1174e-01	1	5.1174e-01
5.1174e-01	1	5.1174e-01
5.1174e-01	1	5.1174e-01
5.1175e-01	1	5.1175e-01
5.1175e-01	1	5.1175e-01
5.1187e-01	1	5.1187e-01
5.1279e-01	1	5.1279e-01
-2.8198e+00+ 1.7317e+01i	0.161	1.7545e+01
-2.8198e+00- 1.7317e+01i	0.161	1.7545e+01
-2.6450e+01+ 4.6776e+01i	0.492	5.3736e+01
-2.6450e+01- 4.6776e+01i	0.492	5.3736e+01
-5.4610e+01	1	-5.4610e+01
-5.4621e+01	1	-5.4621e+01
-5.4650e+01	1	-5.4650e+01
-5.4724e+01	1	-5.4724e+01
-5.4952e+01	1	-5.4952e+01
-5.5897e+01	1	-5.5897e+01
-6.3302e+01	1	-6.3302e+01
-2.4982e+02	1	-2.4982e+02
-3.9645e+02	1	-3.9645e+02
-2.1615e+03	1	-2.1615e+03
-2.3269e+03	1	-2.3269e+03
-8.1917e+03	1	-8.1917e+03
-2.1660e+04	1	-2.1660e+04
-2.5780e+04	1	-2.5780e+04

Table 4.4 - Low-Torque Model Eigenvalues

High Torque Eig	High Torque Damping	High Torque Freq (rad/s)
-4.6849e+04	1	-4.6849e+04
-8.4972e+04	1	-8.4972e+04
-1.2623e+05	1	-1.2623e+05
-1.6286e+05	1	-1.6286e+05
-6.1081e+05	1	-6.1081e+05
-1.6673e+06	1	-1.6673e+06
-3.6715e+06	1	-3.6715e+06
-6.7348e+06	1	-6.7348e+06
-1.0076e+07	1	-1.0076e+07

The Bode plot for θ_1/τ_1 (theta_1 to torque_1) is shown in Figure 4.2, while the Bode plot for θ_2/τ_2 is seen in Figure 4.3.

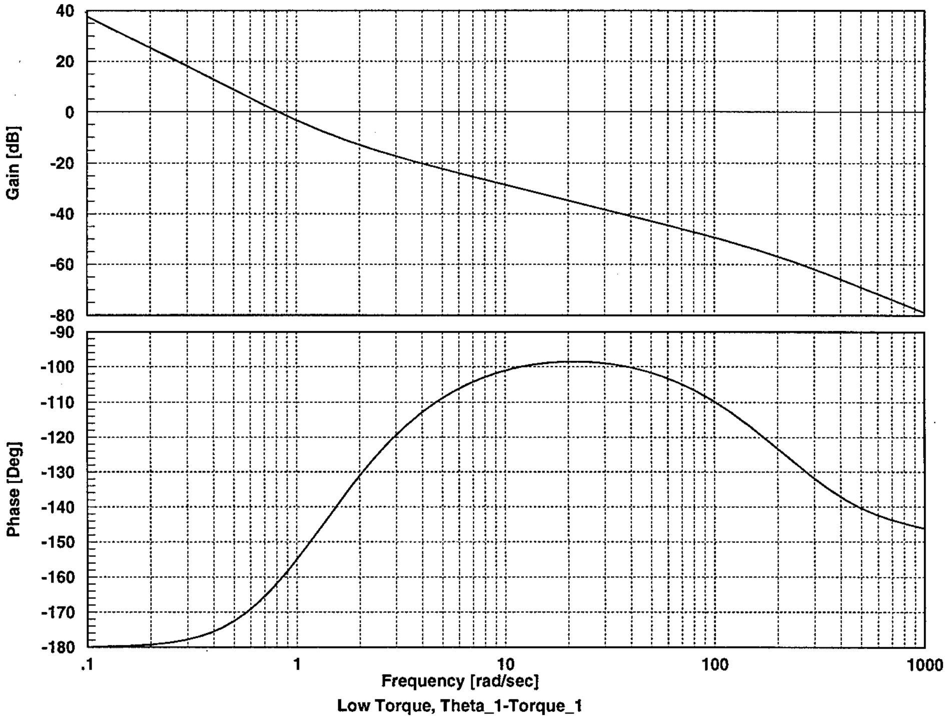


Figure 4.2- Low-Torque Model Bode Plot, θ_1/τ_1

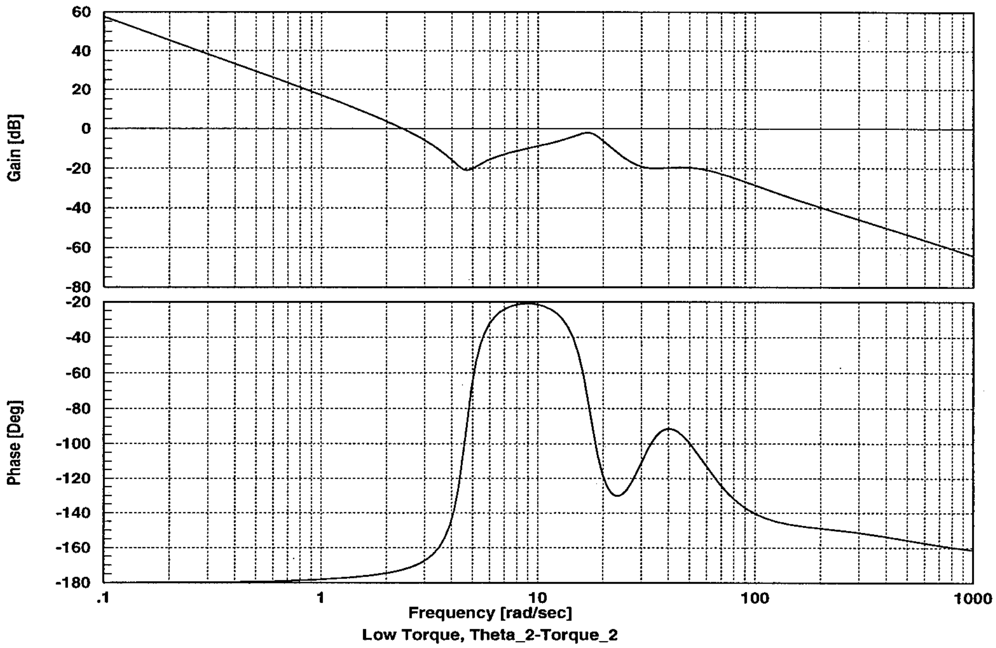


Figure 4.3- Low-Torque Model Bode Plot, θ_2/τ_2

4.5 High-Torque Model

When a higher initial torque onset is applied to the motors, the friction effects are less noticeable. The model is matched to the SANDY controller developed in Chapter 7, and new b and EI values are obtained. Table 4.3 shows these new high-torque values.

Table 4.5 - Experimentally Tuned Values for the High-Torque Model

Link	b	EI
1	.06	5.5
2	.03	5.5

Applying the new values results in a model with eigenvalues shown in Table 4.6. This new model has a Bode plot for θ_1/τ_1 as shown in Figure 4.4. Also, the Bode plot for θ_2/τ_2 is shown in Figure 4.5.

4.6 Model Reduction

Having a 40th order model results in large computational burdens for simulation and the design process. Also, since observer-based output feedback controllers have the same order as the order of the model, it is desirable to decrease the order of the

Table 4.6 - High-Torque Model Eigenvalues

Eig	Damping	Freq (rad/s)
0	1	0
0	1	0
0	1	0
0	1	0
-.846 +/- j 75.1	.011	75.1
-9.3876e-01+/- j5.5952e+01	.017	56.0
-7.9349e+00+/- j2.2995e+02	3.4487e-02	2.3008e+02
-8.1074e+00+/- j1.6425e+02	4.9299e-02	1.6445e+02
-6.8962e+01+/- j6.7478e+02	1.0167e-01	6.7830e+02
-9.6677e+01+/- j5.5960e+02	1.7024e-01	5.6789e+02
-3.5742e+02+/- j1.5023e+03	2.3146e-01	1.5442e+03
-6.1073e+02+/- j1.2901e+03	4.2788e-01	1.4273e+03
-1.2370e+03+/- j2.5928e+03	4.3059e-01	2.8728e+03
-1.7064e+03	1	1.7064e+03
-1.7270e+03	1	1.7270e+03
-1.7834e+03	1	1.7834e+03
-1.9821e+03	1	1.9821e+03
-2.2905e+03+/- j1.5473e+03	8.2864e-01	2.7642e+03
-3.2572e+03+/- j3.3349e+03	6.9872e-01	4.6616e+03
-3.6965e+03	1	3.6965e+03
-3.9464e+03	1	3.9464e+03
-5.4353e+03	1	5.4353e+03
-8.6359e+03	1	8.6359e+03
-1.0523e+04	1	1.0523e+04
-2.1562e+04	1	2.1562e+04
-2.5753e+04	1	2.5753e+04
-3.4189e+04	1	3.4189e+04
-4.8784e+04	1	4.8784e+04
-7.3864e+04	1	7.3864e+04

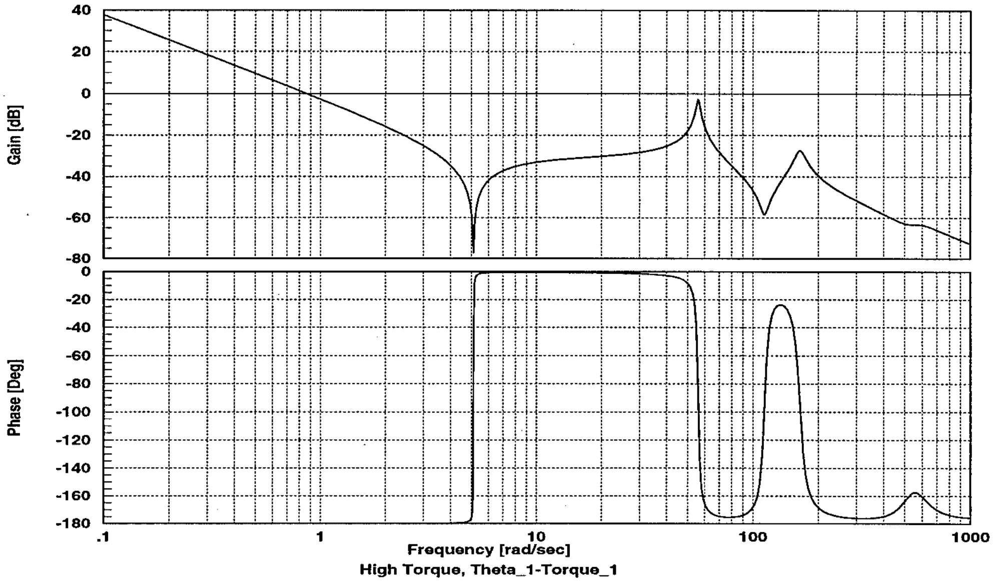


Figure 4.4- High-Torque Model Bode Plot, θ_1/τ_1

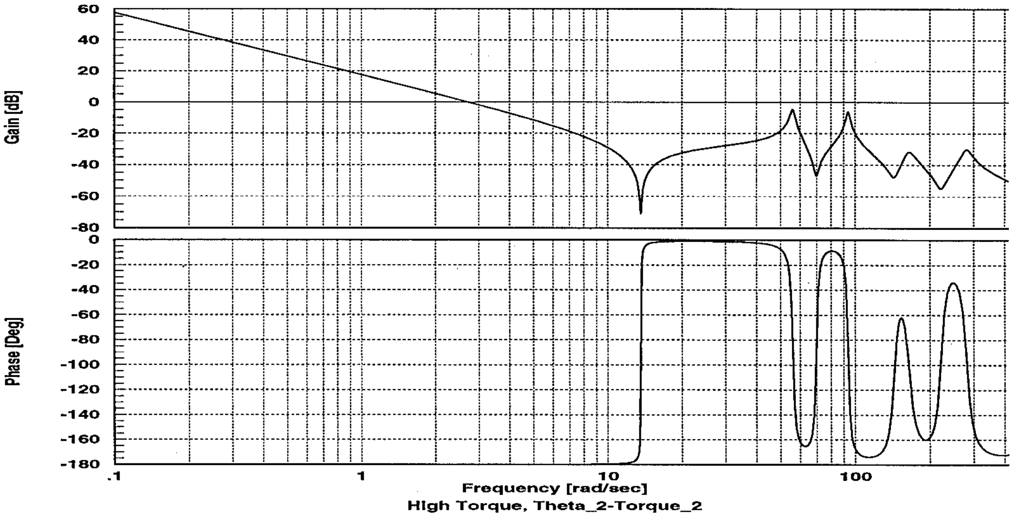


Figure 4.5- High-Torque Model Bode Plot, θ_2/τ_2

plant model as much as possible without losing much dynamic information. One way to reduce the model order is to balance the system using a balanced realization algorithm and removing the non-dominant modes (those with low Hankel singular values). This was accomplished by use of the MATLAB balsq.m command which balances and truncates the system. In order to perform the balancing on a system which is not asymptotically stable, a special algorithm is needed which first removed the 4 rigid body poles at the origin, balanced the remaining part of the system, and then reattached the rigid body modes.

By applying the balanced truncation method, the plant was reduced to 18th order for the high-torque model. Figure 4.6 and Figure 4.7 display the Bode plots for the reduced model for the high-torque model. Notice the reduced model matches the full-order models well below 100 rad/s, which is the primary area of concern for this flexible manipulator. Table 4.8 shows the eigenvalues for the reduced system. Note the close match of the eigenvalues between the reduced-order model and the full-order model eigenvalues.

Table 4.7 - 18th Order Reduced High-Torque Model Eigenvalues

Eigenvalues	Damping	Frequency (rad/s)
-8.6736e-18+ 6.9728e-10i	1.2439e-08	6.9728e-10
-8.6736e-18- 6.9728e-10i	1.2439e-08	6.9728e-10
-1.5408e-14+ 2.0437e-08i	7.5392e-07	2.0437e-08
-1.5408e-14- 2.0437e-08i	7.5392e-07	2.0437e-08
-8.4597e-01+ 7.5120e+01i	1.1261e-02	7.5125e+01
-8.4597e-01- 7.5120e+01i	1.1261e-02	7.5125e+01
-9.3877e-01+ 5.5952e+01i	1.6776e-02	5.5960e+01
-9.3877e-01- 5.5952e+01i	1.6776e-02	5.5960e+01
-7.9390e+00+ 2.2995e+02i	3.4505e-02	2.3009e+02
-7.9390e+00- 2.2995e+02i	3.4505e-02	2.3009e+02
-8.1176e+00+ 1.6425e+02i	4.9361e-02	1.6445e+02
-8.1176e+00- 1.6425e+02i	4.9361e-02	1.6445e+02
-7.0065e+01+ 6.7562e+02i	1.0315e-01	6.7925e+02
-7.0065e+01- 6.7562e+02i	1.0315e-01	6.7925e+02
-1.0330e+02+ 5.6669e+02i	1.7933e-01	5.7603e+02
-1.0330e+02- 5.6669e+02i	1.7933e-01	5.7603e+02
-4.4510e+02+ 1.6607e+03i	2.5889e-01	1.7193e+03
-4.4510e+02- 1.6607e+03i	2.5889e-01	1.7193e+03

Similarly, balanced truncation was applied to the low-torque model. In this case, only a 33rd order reduced model was obtained successfully. Figure 4.8 and Figure

4.9 show the Bode plots for the reduced order system. Notice that it matches the full-order system well. Table 4.8 shows the eigenvalues for the reduced 33rd order system.

Table 4.8 - 33rd Order Reduced Low-Torque Model Eigenvalues

Eigenvalues	Damping	Frequency (rad/s)
8.4504e-08	-1	8.4504e-08
2.1235e-09	-1	2.1235e-09
-2.1235e-09	1	2.1323e-09
-8.4516e-08	1	8.4516e-08
-5.1186e-01	1	5.1186e-01
-5.1301e-01	1	5.1301e-01
-9.3523e-01	1	9.3523e-01
-2.8207e+00+ 1.7319e+01i	1.6075e-01	1.7547e+01
-2.8207e+00- 1.7319e+01i	1.6075e-01	1.7547e+01
-2.6465e+01+ 4.6794e+01i	4.9229e-01	5.3760e+01
-2.6465e+01- 4.6794e+01i	4.9229e-01	5.3760e+01
-5.4620e+01	1	5.4620e+01
-5.4620e+01+ 6.5820e-01i	9.9993e-01	5.4624e+01
-5.4620e+01- 6.5820e-01i	9.9993e-01	5.4624e+01
-5.4648e+01	1	5.4648e+01
-5.4868e+01	1	5.4648e+01
-5.5914e+01	1	5.5914e+01
-6.4390e+01	1	6.4390e+01
-2.4392e+02	1	2.4392e+02
-3.9608e+02	1	3.9608e+02
-2.0611e+03	1	2.0611e+03
-2.3130e+03	1	2.3130e+03
-8.3537e+03	1	8.3537e+03
-2.2131e+04	1	2.2131e+04
-3.2986e+04	1	3.2986e+04
-4.6661e+04	1	4.6661e+04
-8.4488e+04	1	8.4488e+04
-1.2356e+05	1	1.2356e+05
-4.1591e+05+ 7.7674e+04i	9.8300e-01	4.2310e+05
-4.1591e+05- 7.7674e+04i	9.8300e-01	4.2310e+05
-2.0474e+06	1	2.0474e+06
-4.1778e+06	1	4.1778e+06
-6.7552e+06	1	6.7552e+06

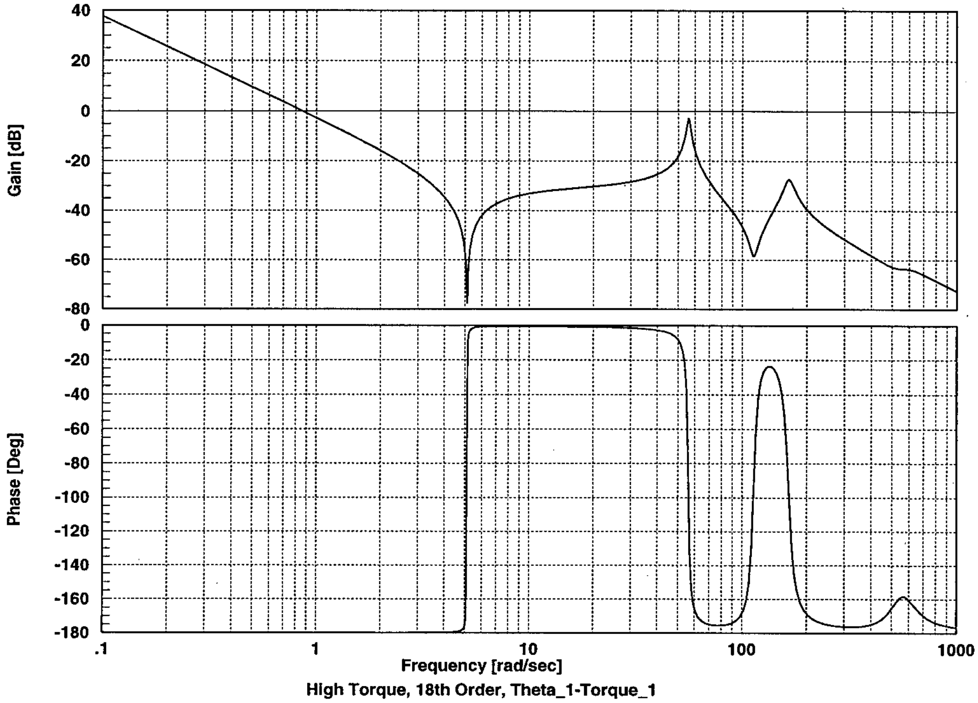


Figure 4.6- High-Torque Model Bode Plot, θ1/τ1 of the Reduced-Order Model

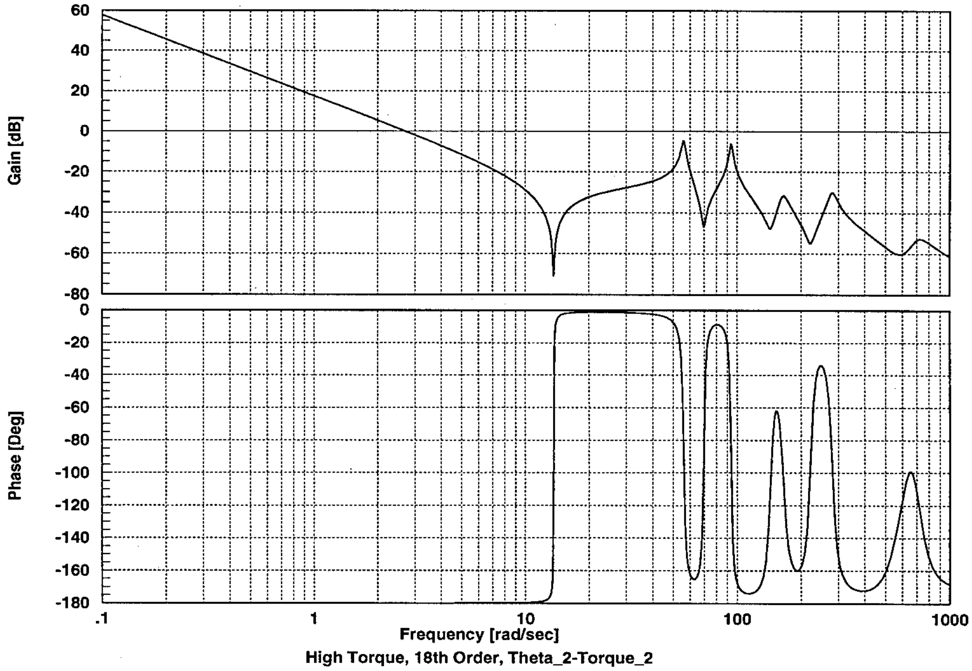


Figure 4.7- High-Torque Model Bode Plot, θ2/τ2 of the Reduced-Order Model

4.7 Experimental Model Validation

One way to determine the validity of a model is to conduct a frequency response of the plant and compare the experimental results to the mathematical model frequen-

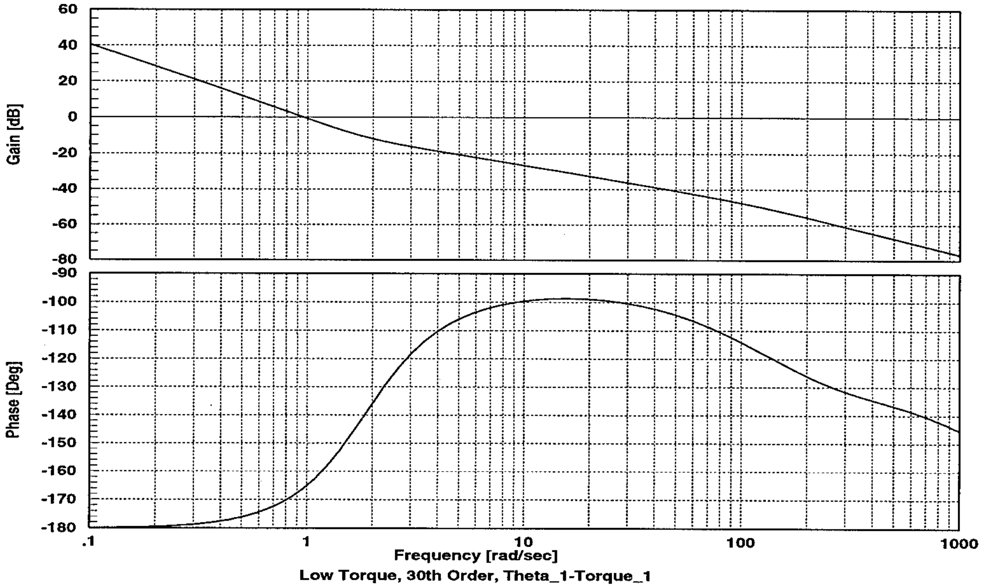


Figure 4.8- Low -Torque Model Bode Plot, θ1/τ1 of the Reduced-Order Model

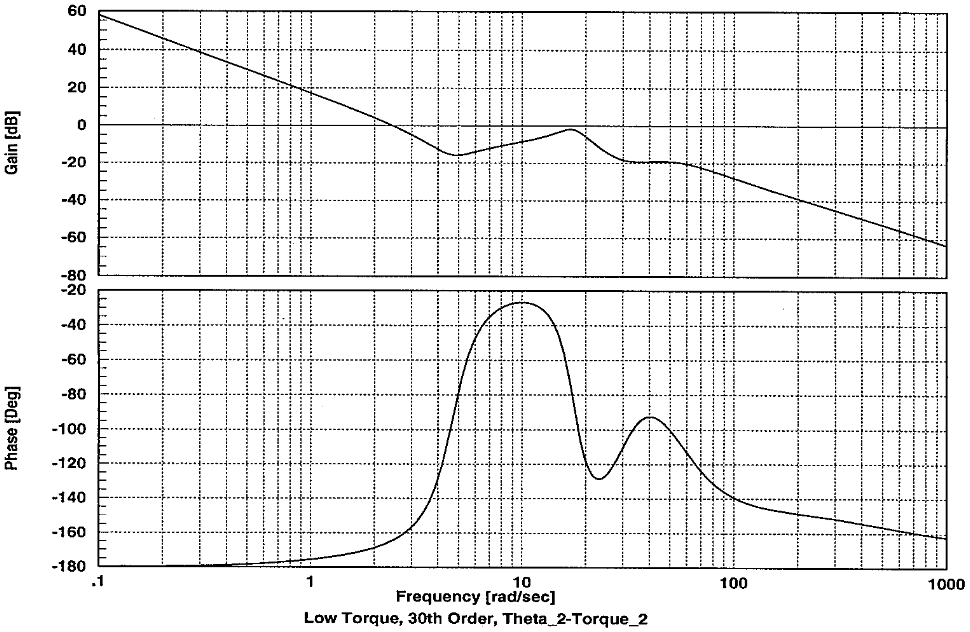


Figure 4.9- Low -Torque Model Bode Plot, θ2/τ2 of the Reduced-Order Model

cy response. Since most of the design work involves the high-torque model, only it will be compared to the experimental data directly.

In order to acquire the frequency data experimentally, a sine wave voltage is fed directly into one of the motors. Ideally, no torque should be applied to the second motor for the most accurate representation of the open-loop model. As a matter of practice, though, a small proportional gain is required to hold the second link in place during sine sweeps of the first. For these small proportional gains, the resulting experimental model is close to the open-loop model [56]. Specifically, a proportional gain of 0.5 is used to close the loop of the motor which is not sinusoidally excited.

To experimentally implement the sine sweep for the second link, the setup of Figure 4.10 is used. This feeds a sinusoidal voltage into one motor while keeping the other motor at a specified reference position. To conduct a sine sweep for the first link, just reverse the reference and sinusoid of Figure 4.10.

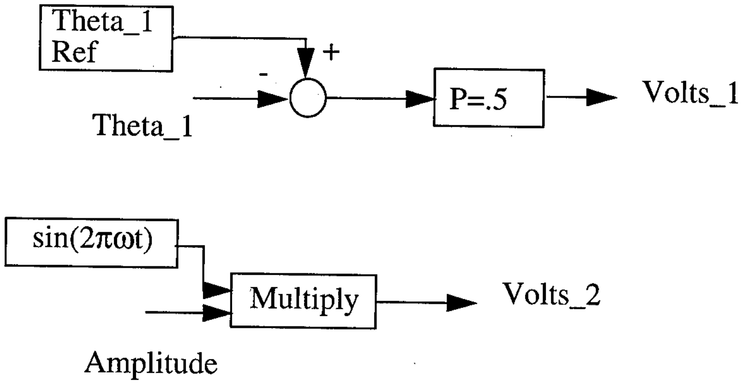


Figure 4.10- Experimental Sine Sweep Block Diagram (Link 2)

The raw data is taken as volts (the input) and theta (the output). Of course, the actual units required are Newton-meters (Nm) of torque for input. Thus, knowing the conversion from Nm to volts for each motor and the amplitude of the response in radians, the transfer function of degrees/Nm for amplitude is obtained. By knowing the frequency and the delay in seconds between the input and output, the phase shift in degrees is obtained. Figure 4.11 shows the experimental vs. model frequency response for the magnitude, while Figure 4.12 shows the corresponding phase plot for the shoulder.

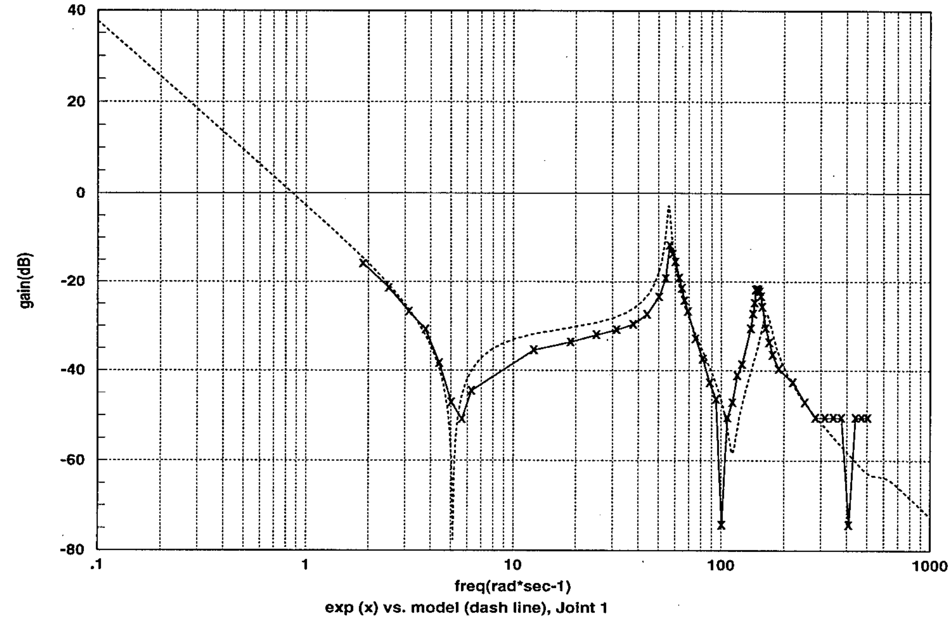


Figure 4.11- Shoulder Magnitude Frequency Response

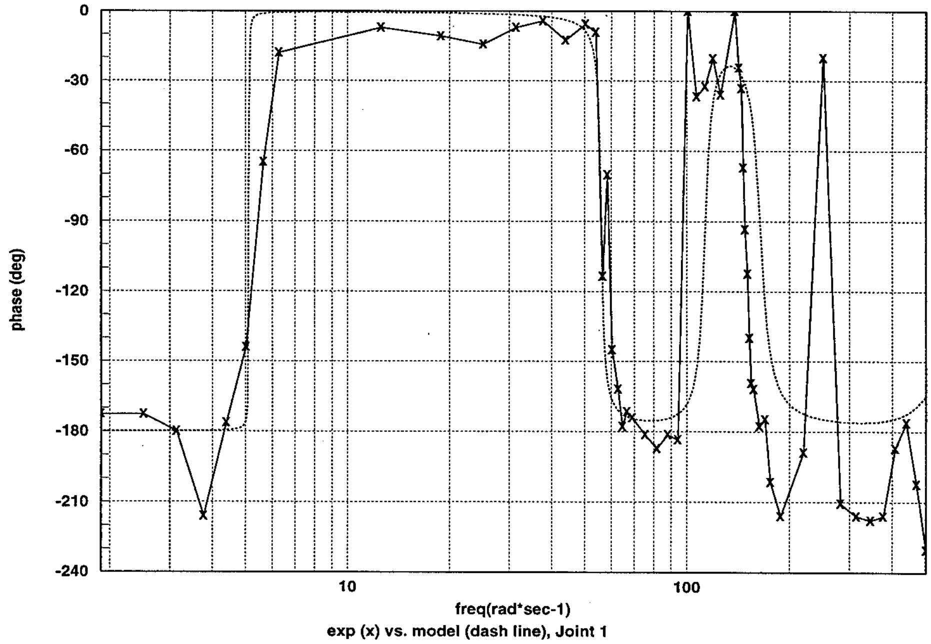


Figure 4.12- Shoulder Phase Frequency Response

The elbow magnitude frequency response is shown in Figure 4.13 while the phase response is shown in Figure 4.14.

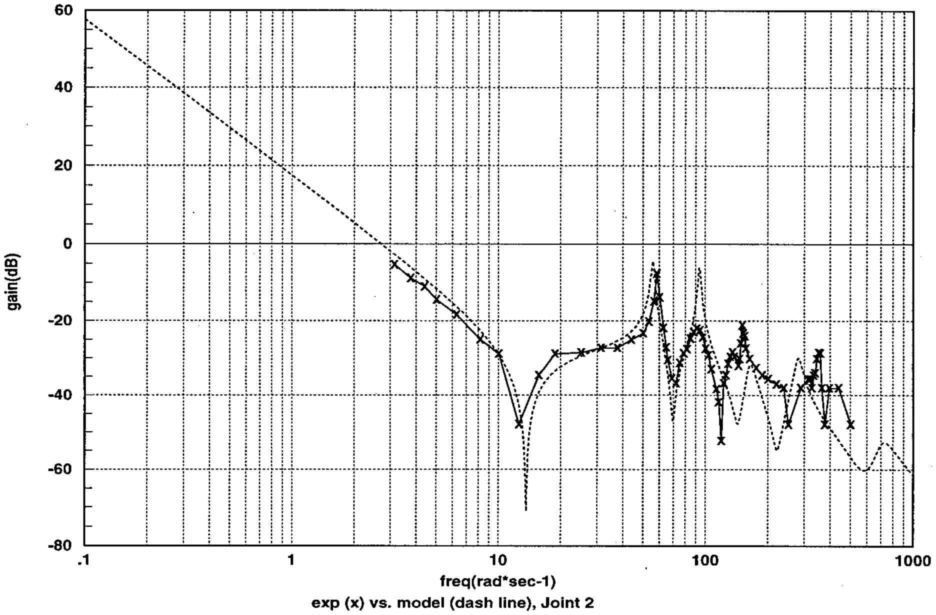


Figure 4.13- Elbow Magnitude Frequency Response

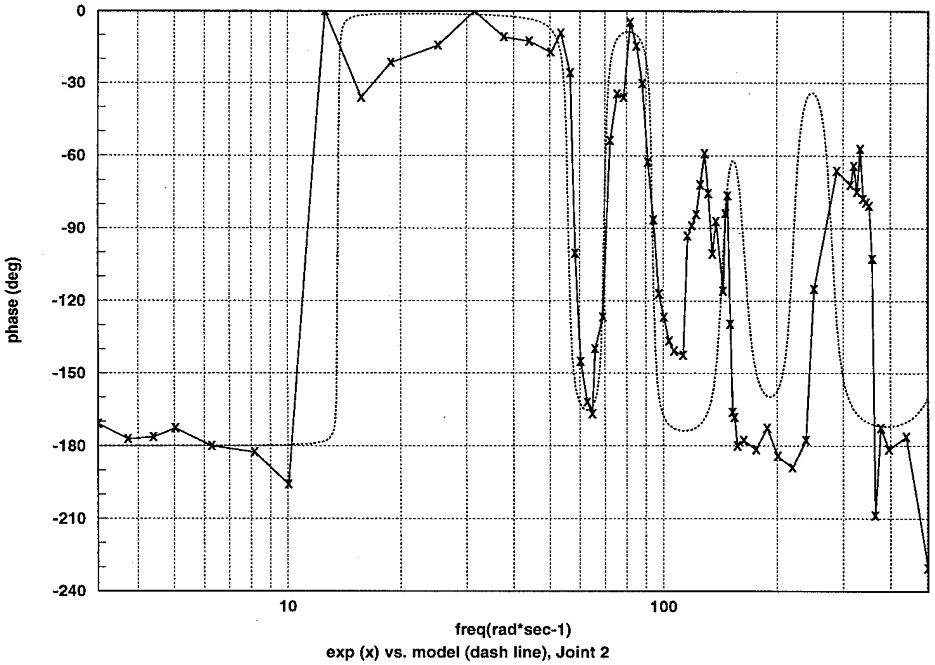


Figure 4.14- Elbow Phase Frequency Response

4.8 Mode Shapes

Each of the peaks on the Bode plot correspond to flexible modes of the arm. The modes alternate between the first and second link, starting with mode 1 on the first link. Thus mode 2 is actually the first mode of the second link. The mode shape can be determined by the number of nodes it has, with the number of nodes equal to the link mode minus one. Thus, the first and second modes are essentially first modes so they have no nodes. The third mode is the second mode for the first link, and therefore has one node.

For the semi-flexible links used in this research, the mode deflections are relatively small and therefore hard to see in a single camera shot. However, careful examination of the pictures can show some of the mode shape. First, Figure 4.15 shows mode 1, which has no nodes and is on the 1st link. There is a slight deviation to the right of the centerline in this picture.

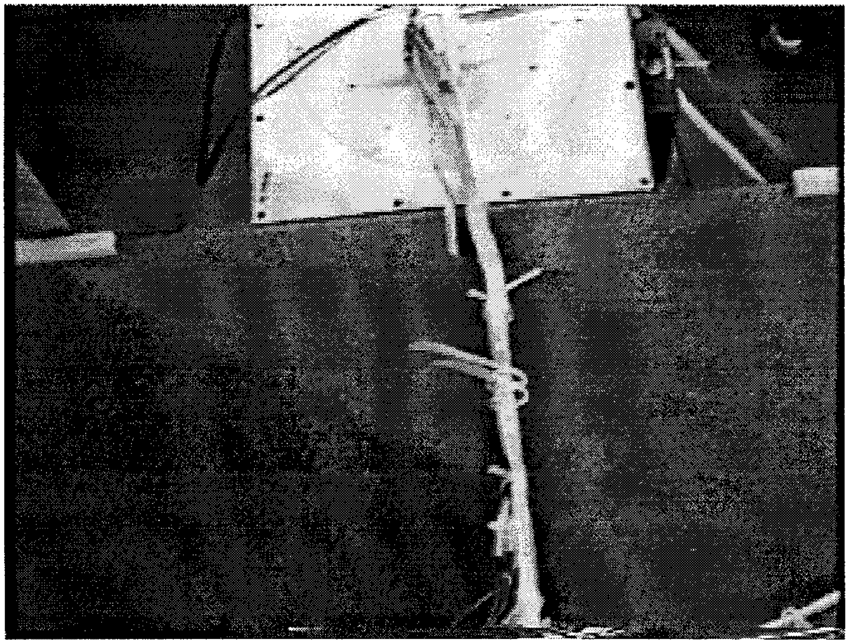


Figure 4.15- 1st Mode, Semi-Flexible Links (Link 1, Mode 1)

The second mode can be seen in Figure 4.16. This is the first mode for the second link and thus has no nodes. There is a slight deviation to the left here, and clearly no nodes.

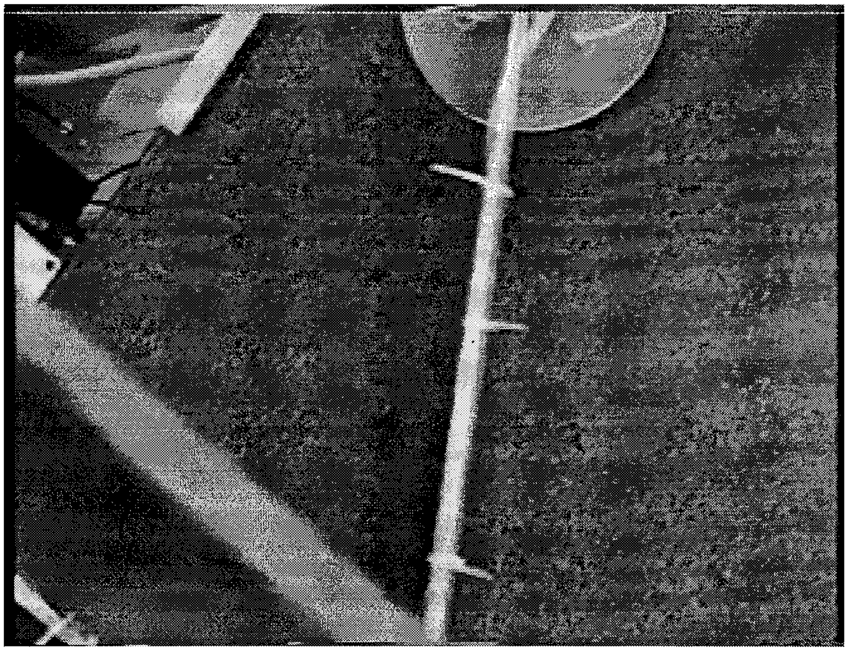


Figure 4.16- 2nd Mode, Semi-Flexible Links (Link 2, Mode 1)

Finally, the third mode is shown in Figure 4.17. Here, a node appears roughly in the center of the first link.

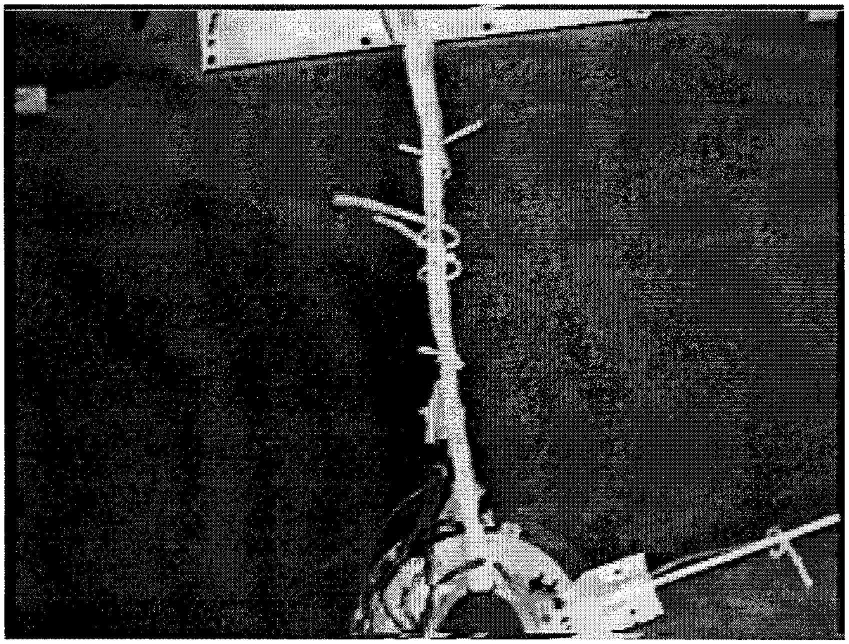


Figure 4.17- 3rd Mode, Semi-Flexible Links (Link 1, Mode 2)

5 PID Design and Experiments

In order to establish a standard for comparison and to experimentally validate the system, simple baseline experiments are performed. In particular, a basic proportional-integral-derivative (PID) controller is simulated and experimentally tested on the two-link arm. Additional plots are shown in Appendix B for different test trajectories and verify that the results for this trajectory are valid for others as well.

5.1 PID Design

Logically, the first control law implemented on a new system should be as simple as possible. Therefore, initial testing is done with a simple proportional-integral-derivative (PID) controller [17]. It is represented with equation (5.1).

$$\tau(s) = \left(K_p + \frac{K_I}{s} + K_D s\right) y(s) \tag{5.1}$$

To achieve better performance, the PID controller was tuned iteratively with hardware runs to achieve good performance in trajectory tracking. Table 5.1 lists the high performance PID gains.

Table 5.1 - High Performance PID Gains

Link	K_P	K_I	K_D
1	1	.5	2
2	1	.5	2

This PID controller was implemented on the AC100 system with a sampling time of 1 ms. Figure 5.1 shows a diagram of the test trajectories listed in Table 5.2.

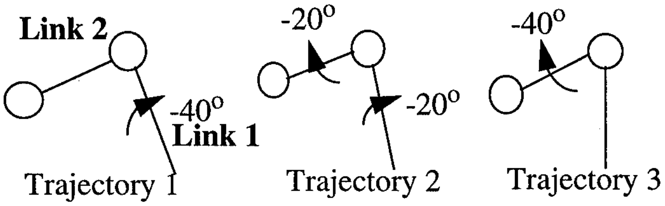


Figure 5.1- TLA Test Trajectories

Figure 5.2 shows the reference commands for trajectory 2 while Figure 5.3 shows the angle conventions.

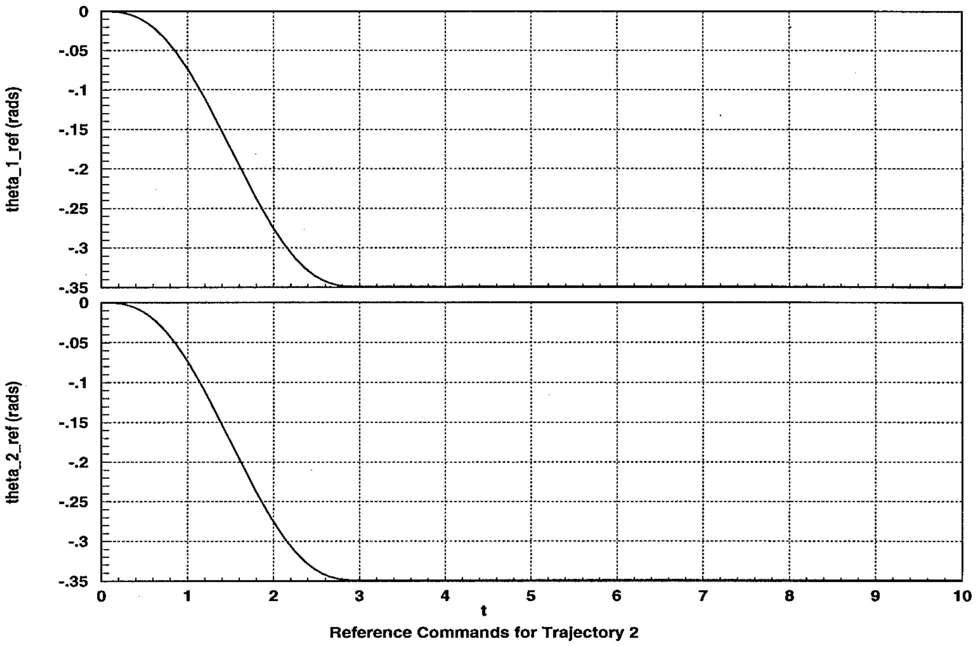


Figure 5.2- Trajectory 2 Reference Commands

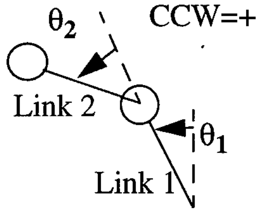


Figure 5.3- Angle Conventions

Table 5.2- Test Slews for Parameter ID

Trajectory	θ1_start (deg)	θ1_end (deg)	θ2_start (deg)	θ2_end (deg)
1	40	0	90	90
2	20	0	110	90
3	0	0	110	70

Figure 5.4 shows the experimental results for trajectory 2. Figure 5.5 and Figure 5.6 show experimental versus simulation results for trajectory 2. Please note that all the plots in this report have time units of seconds and have angle outputs relative to the starting position. Notice the reasonably close match between experimental and simulation for both angles.

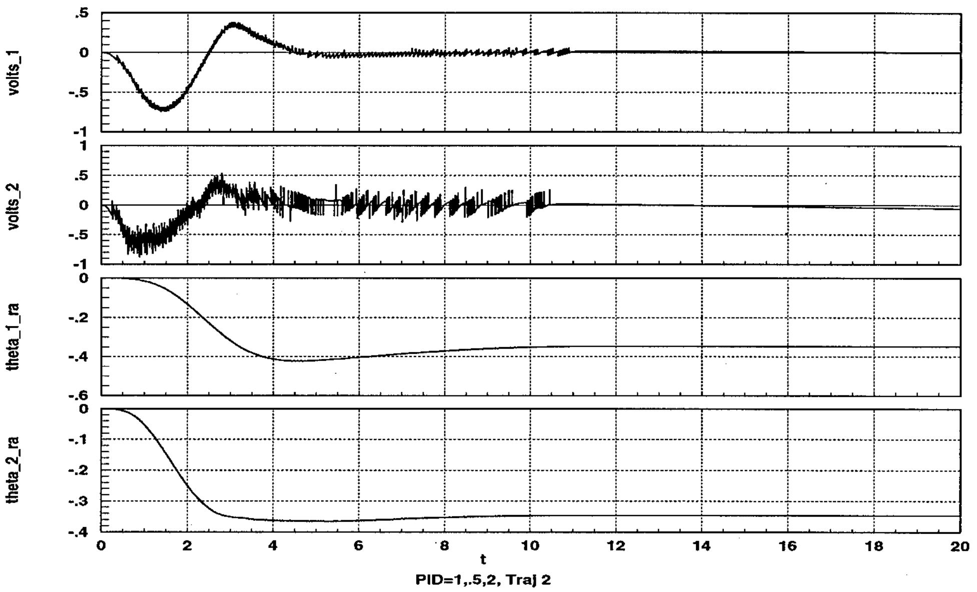


Figure 5.4- PID Experimental Results, Trajectory 2

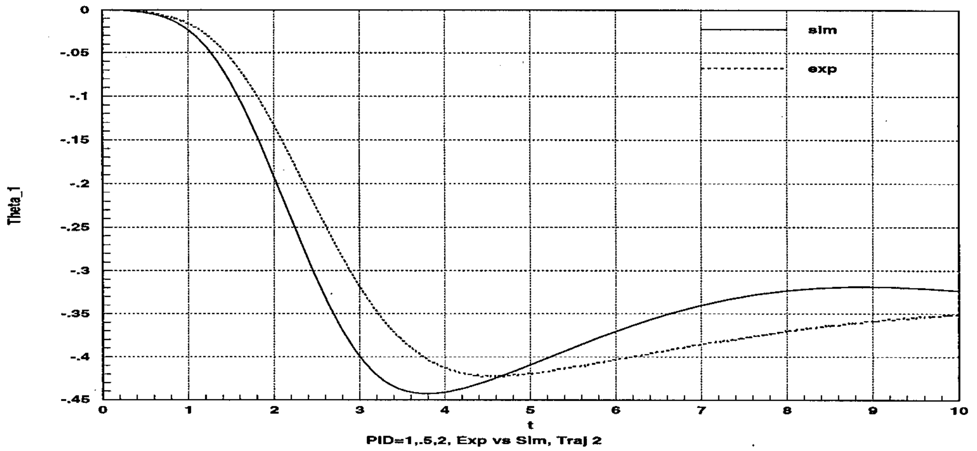


Figure 5.5- PID Experimental vs. Simulation, Trajectory 2, θ_1

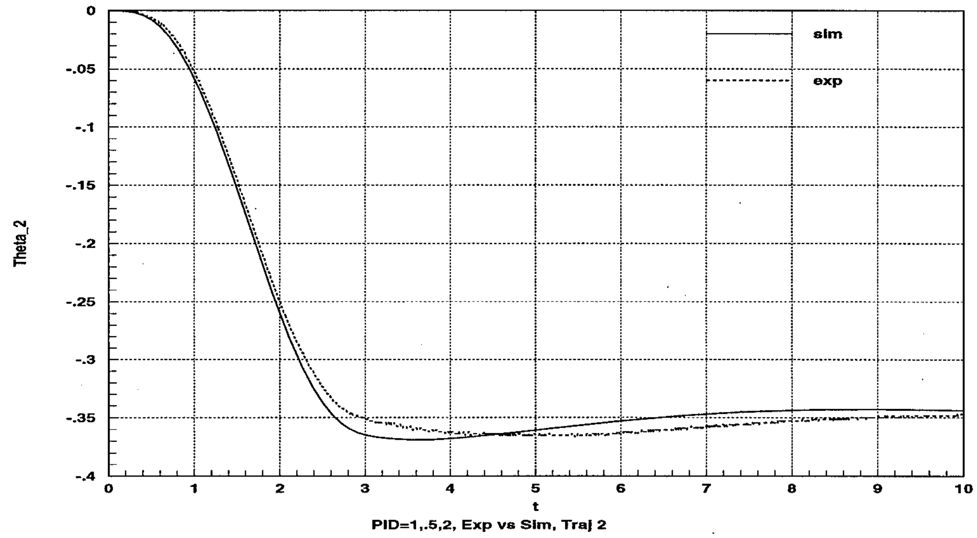


Figure 5.6- PID Experimental vs. Simulation, Trajectory 2, θ_2

5.2 PID Control Bandwidth

Analyzing the closed-loop controller bandwidth gives insight into the physical controller performance. Figure 5.7 shows the θ_1/θ_{1cmd} control bandwidth.

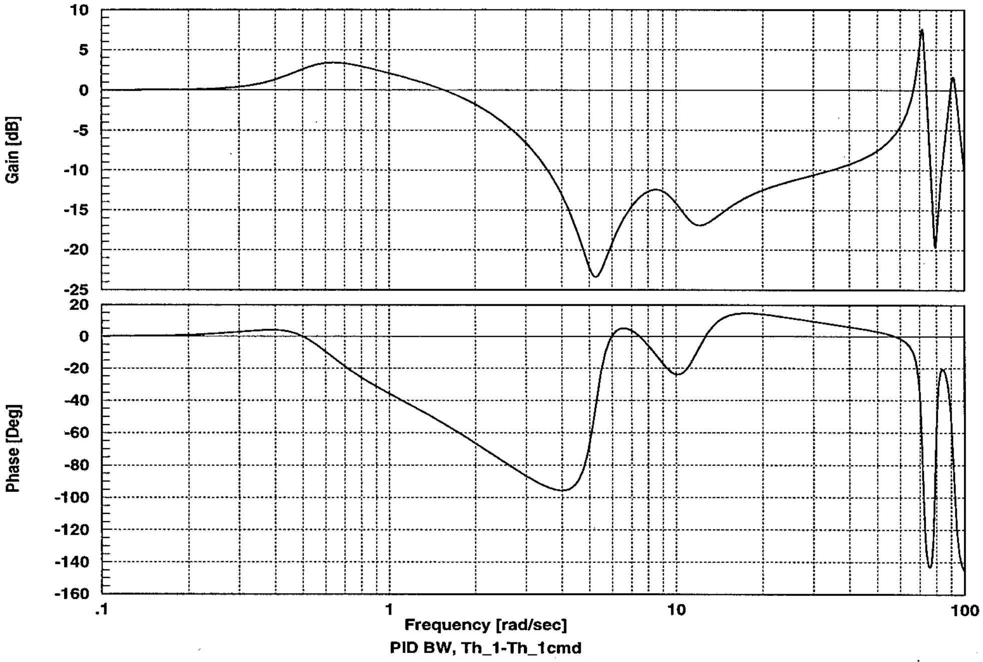


Figure 5.7- PID θ_1/θ_{1cmd} Control Bandwidth

Figure 5.8 shows the θ_2/θ_{2cmd} control bandwidth. The convention for this research is to define the bandwidth where the magnitude first crosses the -3 dB magnitude. Notice the initial bandwidth for θ_{1cmd} is about 2 rad/s, but has a peak resonance at about 70 rad/s. Likewise, the second loop has an initial bandwidth of about 9 rad/s, but has a resonant peak at about 90 rad/s.

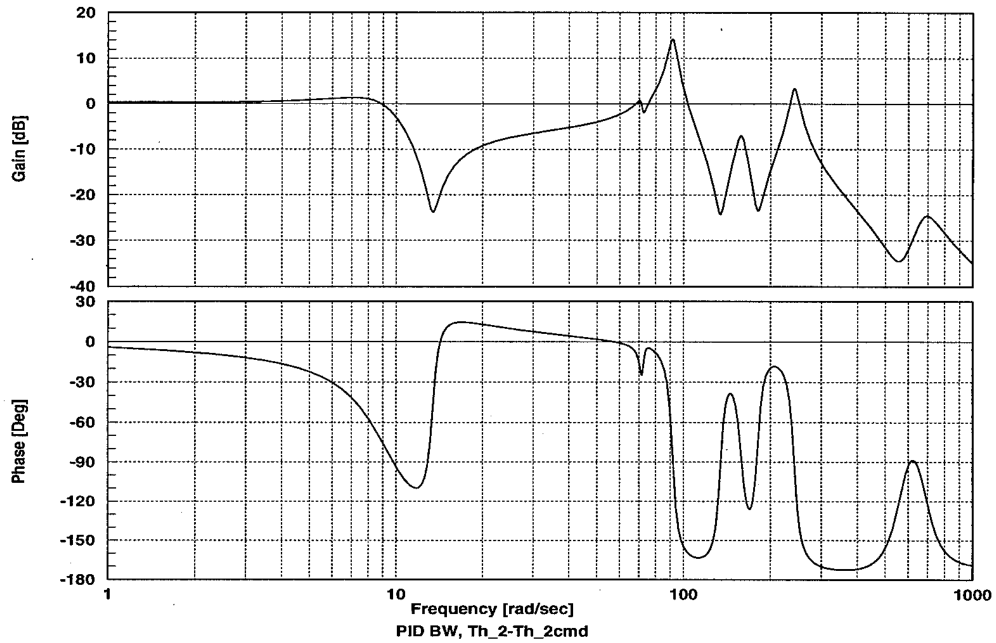


Figure 5.8- PID θ_2/θ_{2cmd} Control Bandwidth

6 LQG-Class Controllers Design and Experiments

As PID is the baseline for classical controllers, Linear-Quadratic-Gaussian (LQG) is the baseline for modern (or multiple-input, multiple-output) controllers. This chapter investigates various designs for LQG-class position controllers, and experimentally evaluates them. Since the ultimate goal of this dissertation is to develop reduced-order controllers, it makes sense to evaluate LQG-class controllers, which are full-order controllers.

First, a standard Linear-Quadratic-Gaussian (LQG) controller is designed and tested, and then is extended with Loop Transfer Recovery (LQG/LTR) controller design. Then, Loop Transfer Recovery based on Special Coordinate Basis (SCB/LTR) analysis of the plant is covered. In order to make the LQG/LTR and LTR controllers more practical, controller reduction is applied using modal residualization and balanced truncation.

6.1 LQG Controller

As a baseline, a Linear-Quadratic-Gaussian (LQG) controller is developed. An LQG controller involves a Linear-Quadratic-Regulator (LQR) for full-state feedback gains, and a Linear-Quadratic-Estimator (LQE) to estimate the states from measured outputs. The resulting controller is the same order as the plant.

6.1.1 LQR Background

The details behind LQG control are well-documented [1] [35]. First, a Linear-Quadratic-Regulator (LQR) design determines the performance based on full-state feedback. The objective of an LQR design is to minimize a cost function through the full-state feedback control **u**(t). Given a state space system of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{6.1}$$

The LQR cost function is:

$$J(x_o, u(t), t_f) = \left(\frac{1}{2}\right)\int_{t_o}^{\infty}\left(\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t)\right)dt \tag{6.2}$$

where

- **x**(t) is the state vector

- **u**(t) is the control vector
- **Q** is the penalty on the states
- **R** is the penalty on the control

The control which minimizes the cost function is

$$\boldsymbol{u}(t) = -\boldsymbol{R}^{-1}\boldsymbol{B}^{T}\boldsymbol{S}\boldsymbol{x}(t) \tag{6.3}$$

where **S** is the solution to the steady state Algebraic Riccati Equation:

$$0 = -\boldsymbol{A}^{T}\boldsymbol{S} - \boldsymbol{S}\boldsymbol{A} + \boldsymbol{S}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{T}\boldsymbol{S} - \boldsymbol{Q} \tag{6.4}$$

Often, the optimal gain **G** is represented as

$$\boldsymbol{G} = \boldsymbol{R}^{-1}\boldsymbol{B}^{T}\boldsymbol{S} \tag{6.5}$$

It is also possible to penalize criterion variables in the cost function instead of just the states, even though all the states are still required for full-state feedback. In this case the criterion vector $\mathbf{y}_c$(t) replaces the state vector **x**(t) in the cost function of Equation (6.2).

6.1.2 LQR Application

LQR provides the performance aspect of the LQG controller, so an LQR controller must be designed first. For the TLA, it is desirable to have an integral state in the controller for each control loop to reduce steady-state error. Thus, the 18th order model is augmented with two integral states to produce a 20th order model. The criterion variables are the integral errors for joint 1 and joint 2, as well as $\dot{\theta}_1$ and $\dot{\theta}_2$. After several iterations, the following were the values chosen for **Q** and **R**:

$$\boldsymbol{Q} = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.002 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 70 \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.0001 \end{bmatrix} \tag{6.6}$$

Of course, since the states in the reduced model have no direct physical meaning, full-state feedback cannot be applied to the arm. Thus, some method of state estimation is needed to implement this controller. The optimal gain **G** produced from these values is found in Table 6.1. This gain can be further divided into the estimator gains, **Ge**, and the integral gains, **Gi**. The estimator gains correspond to the first 18 (non-integral) states, and the integral gains correspond to the last 2 (integral) states.

Table 6.1 - LQR Optimal Gain G

State	Torque_1	Torque_2
x_1	-1.2694e+03	1.1974e+03
x_2	1.1818e+03	1.1796e+02
x_3	9.2732e+00	9.0177e+01
x_4	5.8135e+00	-2.6422e+00
x_5	1.5482e+00	4.0738e+00
x_6	7.5760e+00	1.3725e+01
x_7	-2.9797e-01	-9.6501e+00
x_8	-5.0249e-01	3.0444e+00
x_9	-1.3418e+00	1.5026e+01
x_{10}	-1.2585e+00	-5.1786e+00
x_{11}	-5.0386e-02	1.5479e+00
x_{12}	-8.7266e+00	7.9195e+00
x_{13}	-2.4347e-01	-6.8467e+00
x_{14}	2.0838e-02	-3.0895e+00
x_{15}	-3.0416e-02	-1.1861e+00
x_{16}	8.9189e-02	2.1206e+00
x_{17}	5.4804e-01	2.9006e-01
x_{18}	1.1500e-01	-9.4008e-01
x_{19}	-2.2360e+02	1.5767e+00
x_{20}	-1.8656e+00	-2.6457e+02

6.1.3 LQE Background

The Linear-Quadratic-Estimator (LQE) allows implementation of the LQR full-state feedback gains by estimating the states from the sensor outputs. Given the following system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{Ax}(t) + \boldsymbol{Bu}(t) + \Gamma \boldsymbol{w}(t) \tag{6.7}$$

$$\boldsymbol{y}(t) = \boldsymbol{Cx}(t) + \boldsymbol{Du}(t) + \boldsymbol{v}(t) \tag{6.8}$$

where

- $\mathbf{w}(t)$ is a process noise vector, $E[\mathbf{w}(t)]=0$, $E[\mathbf{w}(t+\tau)\mathbf{w}^T(t)]=\mathbf{W}_0\delta(\tau)$
- $\mathbf{v}(t)$ is a sensor noise vector, $E[\mathbf{v}(t)]=0$, $E[\mathbf{v}(t+\tau)\mathbf{v}^T(t)]=\mathbf{V}_0\delta(\tau)$

The problem of optimal linear state estimation is to find the best estimate for the states $\hat{\boldsymbol{x}}(t)$ such that

$$J = \lim_{t \to \infty} E\left[\{ \hat{x}(t) - x(t) \}^T Q \{ \hat{x}(t) - x(t) \} \right] \tag{6.9}$$

The optimal estimator gain matrix **K** which minimizes Equation (6.9) is

$$K = \tilde{X}_{\infty} C^T V_o^{-1} \tag{6.10}$$

where $\tilde{X}_{\infty}$ is the optimal state error covariance which is the solution to the steady-state Algebraic Riccati Equation

$$A\tilde{X}_{\infty} + \tilde{X}_{\infty}A^T - \tilde{X}_{\infty}C^T V_o^{-1} C\tilde{X}_{\infty} + \Gamma W_o \Gamma^T = 0 \tag{6.11}$$

6.1.4 LQE Application

Since the two integral states are part of the control law, they shouldn't be estimated. Therefore, only the other 18 states need to be estimated. One drawback of applying LQE is having to define the process and sensor noise characteristics. In this case, it was decided to have no true process noise and assume the noise was in the sensors. To put some weighting on the plant, a low value of process noise was applied to the control inputs (like LQG/LTR). The LQG controller was designed with the following values:

$$W_o = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, V_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.12}$$

These values produce the **K** gains listed in Table 6.2. The simulation between the LQR and LQG controllers using this **K** are exactly the same, since the initial estimate of the states in the simulation is the same as the actual states.

Table 6.2 - LQE Optimal K (LQG)

Estimated State	y1	y2
x_1	7.5252e-04	-1.6287e-02
x_2	2.6009e-02	-3.7294e-02
x_3	-1.1949e-01	8.5349e-01
x_4	-3.6048e-01	1.0828e+00
x_5	5.0741e-02	6.8298e-02
x_6	-2.3098e-03	1.5583e-03
x_7	4.9844e-04	1.0644e-03

Table 6.2 - LQE Optimal K (LQG)

Estimated State	y1	y2
x_8	-8.4686e-04	2.6056e-02
x_9	7.7974e-04	2.5519e-03
x_{10}	-8.4166e-04	6.4299e-03
x_{11}	-6.9653e-03	4.4648e-03
x_{12}	-6.2393e-04	-2.0622e-04
x_{13}	1.6271e-05	-4.9665e-05
x_{14}	2.1410e-05	3.6260e-04
x_{15}	-2.8443e-07	-2.2607e-06
x_{16}	-8.3606e-06	-1.2918e-05
x_{17}	-1.7028e-05	-1.9769e-06
x_{18}	-1.8682e-05	2.0152e-06

The LQG controller is formed by recombining the estimated states with the two integral states. By defining **Ae=A**(1:18,1:18), **Be=B**(1:18,1:2) and **Ce=C**(3:4,1:18) to correspond to the 18 non-integral states, 2 torque inputs, and 2 integral error outputs, then the controllers are formed with Equations (6.13) - (6.16).

$$\boldsymbol{Acont} = \begin{bmatrix} (\boldsymbol{Ae} - \boldsymbol{KCe} - \boldsymbol{BeGe}) & -\boldsymbol{BeGi} \\ 0 & 0 \end{bmatrix} \tag{6.13}$$

$$\boldsymbol{Bcont} = \begin{bmatrix} \boldsymbol{K}, \mathbf{0}_{18x2} \\ -\boldsymbol{H}, \boldsymbol{I} \end{bmatrix} \tag{6.14}$$

$$\boldsymbol{Ccont} = -\boldsymbol{G} \tag{6.15}$$

$$\boldsymbol{Dcont} = 0_{2x4} \tag{6.16}$$

where **H** is the integral selection matrix and **I** is a 2x2 identity matrix. In this case, **H** is represented by Equation (6.17).

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.17}$$

6.1.5 LQG Implementation

The experimental results of all the controllers are evaluated on one or more of the three trajectories mentioned in Chapter 5. These slews are fed in as command references through a quintic spline for each angle. A picture of the slews and command trajectories are shown in Chapter 5.

The controllers are implemented as digital controllers at a 1 ms sampling time through a Tustin transformation [28]. The initial assumption that digital effects would be small at a 1 ms sampling time was validated through simulation and experimentation. Figure 6.1 shows the experimental data for the LQG controller run at a 1 ms sampling time along trajectory 2. Figure 6.2 and Figure 6.3 show experimental vs. simulation for the same trajectory. Please note that all plots in this chapter have time units of seconds and have angle outputs measured relative to the starting position

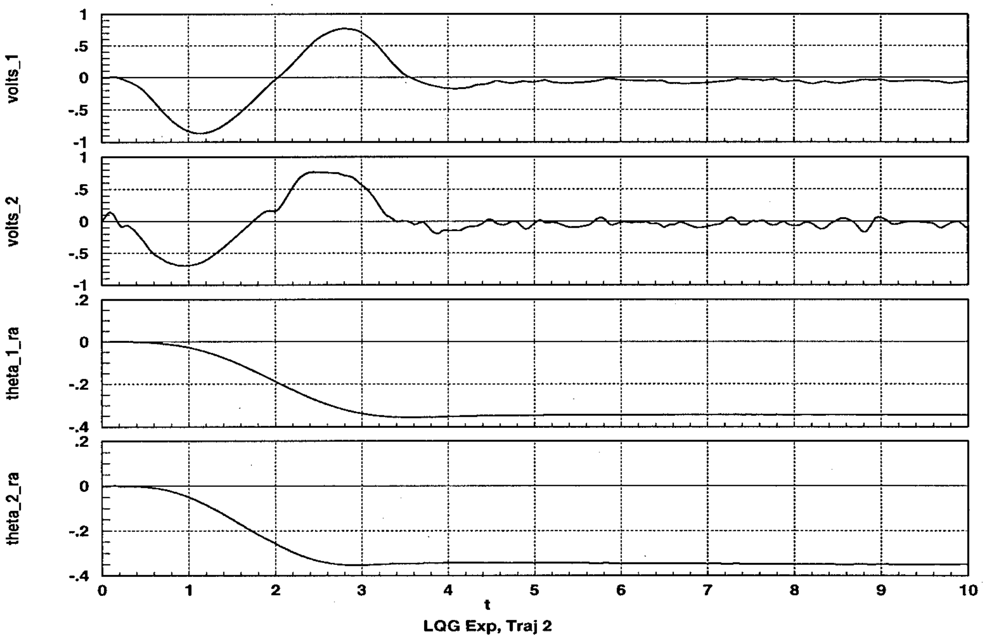


Figure 6.1- LQG Controller Experimental Run - Traj 2

6.2 LQG/LTR

Although LQG implements an LQR design and is guaranteed to be stable, it has no guaranteed robustness. An approach was developed by Doyle [20] to recover robustness asymptotically. This approach involves adding process noise to the control inputs and enables LQR robustness to be recovered if the system is left-invertible and minimum-phase as the noise is increased to infinity. As a practical limit, the designer can't specify infinite noise because the control gains become too large.

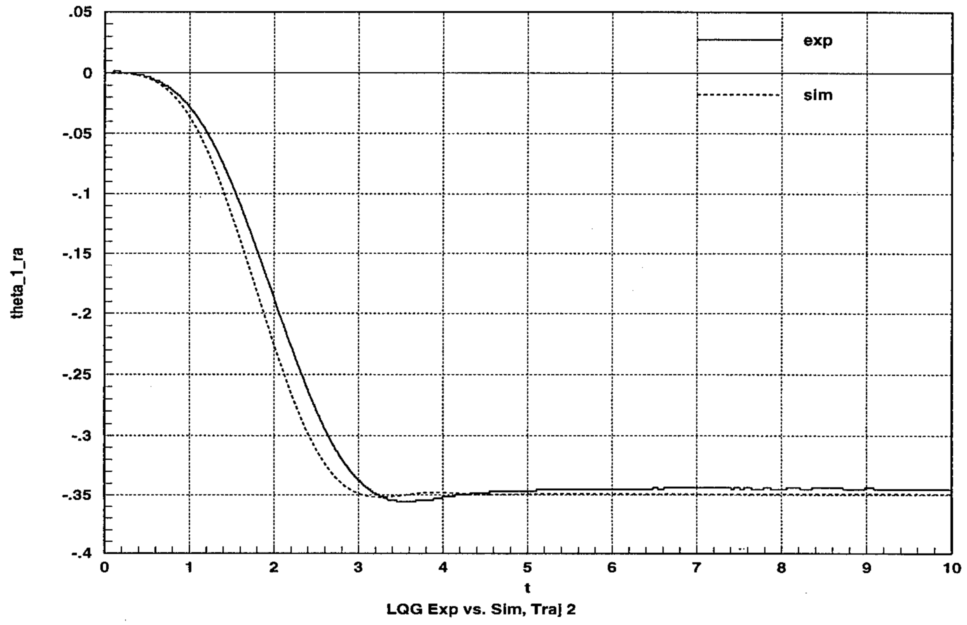


Figure 6.2- LQG, Exp vs. Sim, Traj 2, θ_1

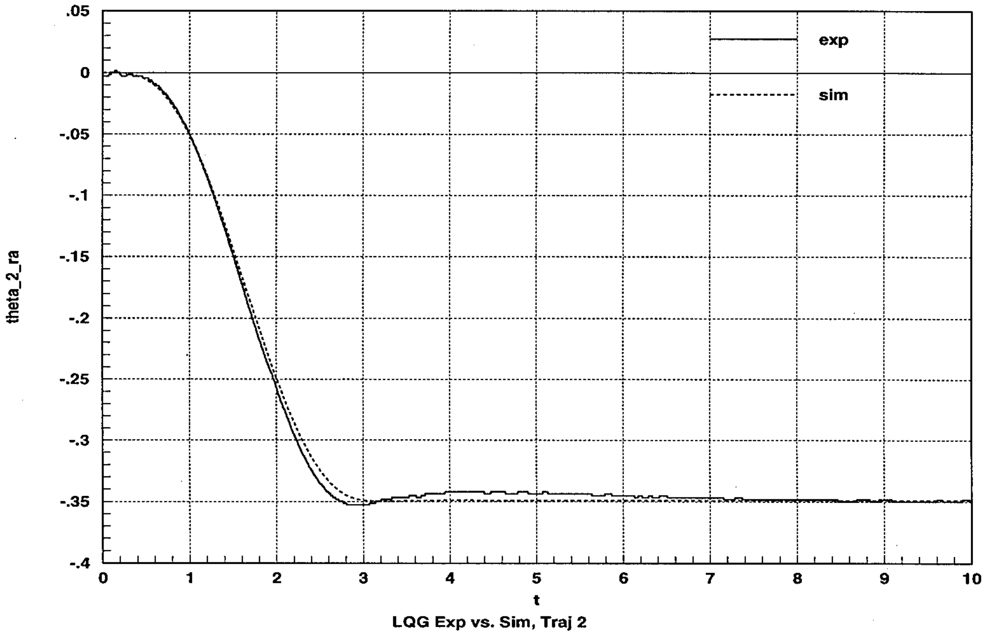


Figure 6.3- LQG, Exp vs. Sim, Traj 2, θ_2

For this application, when LTR process noise weightings were much above 20, the arm would start to vibrate due to numerical roundoff of the finite single-precision word length in the data acquisition system.

The following weightings produced an acceptable LQG/LTR design:

$$W_o = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}, V_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.18}$$

Table 6.2 shows the resulting LQG/LTR estimator gain matrix **K**. Notice that the effect of the LQG/LTR in comparison to the LQG is to provide higher gains in **K**, and thus higher gains in the controller.

Table 6.3 - LQG/LTR Optimal K (LQG)

Estimated State	y1	y2
x_1	1.9298e-01	-2.1687e+00
x_2	4.8338e-01	-2.9498e+00
x_3	-4.2609e+00	3.8446e+01
x_4	-1.4864e+01	4.8814e+01
x_5	2.9153e+01	3.9547e+01
x_6	1.8499e+00	9.8563e+00
x_7	9.5589e-01	-2.9224e+00
x_8	4.1687e-01	2.8492e+01
x_9	1.2974e+00	5.1211e+00
x_{10}	-1.5633e+00	1.2459e+01
x_{11}	-1.3231e+01	8.3106e+00
x_{12}	-1.4522e+00	-8.3386e-03
x_{13}	9.4262e-02	5.9582e-02
x_{14}	5.8507e-02	6.4868e-01
x_{15}	1.0669e-02	4.2726e-02
x_{16}	-2.8636e-02	-8.4883e-03
x_{17}	-8.7449e-02	-5.9608e-02
x_{18}	1.9548e-02	6.6394e-02

The experimental results for the LQG/LTR controller are shown in Figure 6.4 for trajectory 2. The results are very close to the original LQG controller.

6.3 Special Coordinate Basis Loop Transfer Recovery (SCB/LTR)

Another way to recover robustness makes use of special coordinate basis (SCB) analysis. SCB analysis is a method for examining system characteristics such as invertibility and invariant zeros directly from the SCB structure [64]. The Loop Transfer Recovery uses the SCB form to minimize the difference between the observer loop and the LQR loop [62] [63].

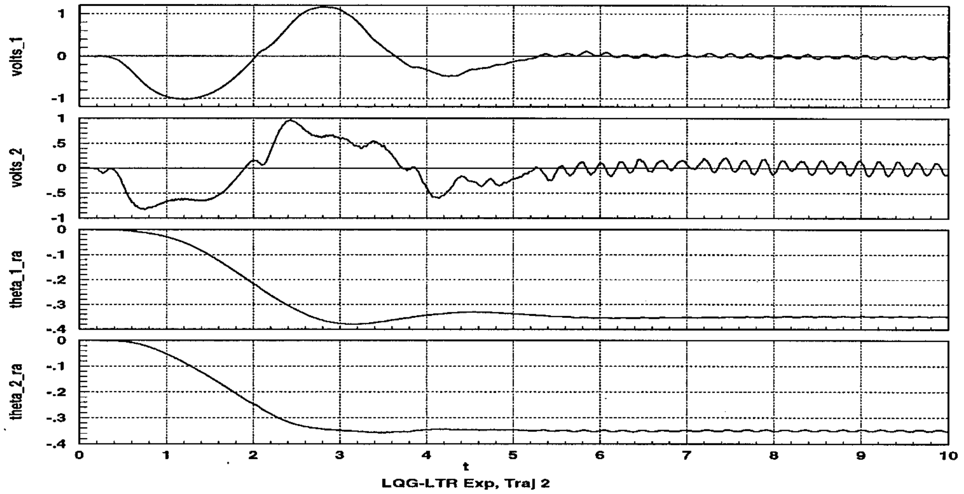


Figure 6.4- LQG/LTR Controller Experimental Run - Traj 2

6.3.1 SCB Background

Given a system of the form

$$\dot{\hat{\boldsymbol{x}}}(t) = \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{B}}\boldsymbol{u}(t) \tag{6.19}$$

$$\hat{\boldsymbol{y}}(t) = \hat{\boldsymbol{C}}\hat{\boldsymbol{x}}(t) \tag{6.20}$$

SCB transforms this system using 3 transformations. Γ_1 transforms the state, Γ_2 transforms the output, and Γ_3 transforms the control of the original system to SCB form. Equations show the transformations to SCB form.

$$\hat{\boldsymbol{x}} = \Gamma_1 \begin{bmatrix} \boldsymbol{x}_a \\ \boldsymbol{x}_b \\ \boldsymbol{x}_c \\ \boldsymbol{x}_f \end{bmatrix} \tag{6.21}$$

$$\hat{\boldsymbol{y}} = \Gamma_2 \begin{bmatrix} \boldsymbol{y}_f \\ \boldsymbol{y}_s \end{bmatrix} \tag{6.22}$$

$$\hat{u} = \Gamma_3 \begin{bmatrix} u \\ v \end{bmatrix} \tag{6.23}$$

where

- $\mathbf{x}_a$ are the states not coupled directly to the input or output
- $\mathbf{x}_b$ are the states not directly affected by the input but are output as $\mathbf{y}_s$
- $\mathbf{x}_c$ are the states affected by the input $\mathbf{v}$ but are not directly output
- $\mathbf{x}_f$ are the states directly affected by the input $\mathbf{u}$ and affect output $\mathbf{y}_f$
- $\mathbf{y}_f$ are the fast outputs
- $\mathbf{y}_s$ are the slow outputs
- $\mathbf{u}$ and $\mathbf{v}$ are the SCB inputs

According to SCB, the size of the $\mathbf{x}_a$, $\mathbf{x}_b$, $\mathbf{x}_c$, $\mathbf{x}_f$ vectors determine several important system characteristics. Namely, the size of $\mathbf{x}_a$ determines the number of system invariant zeros. Further, $\mathbf{x}_a$ can be divided into 3 subvectors, $\mathbf{x}_{a-}$, $\mathbf{x}_{a0}$, and $\mathbf{x}_{a+}$. The number of invariant zeros in the left-half plane correspond to the dimension of $\mathbf{x}_{a-}$. Similarly, the number of invariant zeros on the imaginary axis is equal to the dimension of $\mathbf{x}_{a0}$. Finally, the number of non-minimum-phase zeros is equal to the dimension of $\mathbf{x}_{a+}$.

Right-invertibility of a system is determined by the dimension of $\mathbf{x}_b$. If $\mathbf{x}_b$ has a dimension of 0, then the system is right-invertible. Similarly, left-invertability is determined in the same manner by the dimension of $\mathbf{x}_c$. Lastly, the number of infinite zeros is equal to the number of fast states, $\mathbf{x}_f$.

Another important feature of SCB decomposition is that the invariant zeros can be calculated. The SCB $\mathbf{A}$ matrix has a sub-matrix $\mathbf{A}_{aa}$ associated with the states of $\mathbf{x}_a$. As it turns out, the eigenvalues of the $\mathbf{A}_{aa}$ submatrix are the invariant zeros of the original system.

6.3.2 SCB Plant Analysis

An SCB analysis was applied to the original 40th order arm model for analysis purposes. When performing an open-loop analysis, the MATLAB tzero.m command stated that there were no invariant zeros. Obviously, this was an error. Using the SCB toolbox [41], the 40th order system was discovered to have 36 minimum-phase invariant zeros and 4 infinite zeros.

As stated earlier, all the controllers in this paper used the 18th order model, so an analysis was performed on this model. The reduced model possessed 14 minimum-phase zeros, 2 large non-minimum-phase zeros, and 2 infinite zeros. Table 6.4 lists the 18th order invariant zeros. The non-minimum phase zeros are so large that they are essentially infinite zeros and shouldn't limit the bandwidth of the designed controllers.

Table 6.4 - 18th Order Model Invariant Zeros

Invariant Zeros
1.8641e+05+ 1.5797e+05i
1.8641e+05- 1.5797e+05i
-3.6493e+02+ 1.5033e+03i
-3.6493e+02- 1.5033e+03i
-5.3162e+01+ 5.6874e+02i
-5.3162e+01- 5.6874e+02i
-7.3953e+01+ 4.9774e+02i
-7.3953e+01- 4.9774e+02i
-4.9655e+00+ 1.7204e+02i
-4.9655e+00- 1.7204e+02i
-1.3772e+00+ 8.1386e+01i
-1.3772e+00- 8.1386e+01i
-6.8069e-02+ 1.7224e+01
-6.8069e-02- 1.7224e+01i
-8.7010e-03+ 5.4323e+00i
-8.7010e-03- 5.4323e+00i

6.3.3 SCB/LTR Background

Loop Transfer Recovery based on SCB form has the advantage that it directly minimizes the error between the loop transfer function of the observer and the target loop of the LQR design (and its desirable robustness). Figure 6.5 shows a block diagram of the target LQR loop.

The corresponding state feedback target loop is given in Equation (6.24).

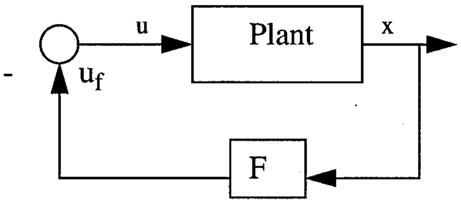
$$\boldsymbol{L}_t(s) = \boldsymbol{F}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B} \tag{6.24}$$

In contrast, Figure 6.6 shows an output feedback loop. Here, **y** is the output given by Equation (6.25).

$$\boldsymbol{y}(t) = \boldsymbol{Cx}(t) + \boldsymbol{Du}(t) \tag{6.25}$$

The corresponding loop equation is given in Equation (6.26).

$$\boldsymbol{L}(s) = \boldsymbol{C}(s)\boldsymbol{P}(s) \tag{6.26}$$



Plant :

$\dot{x}$=Ax+Bu

u_f=Fx

Figure 6.5- LQR Target Loop

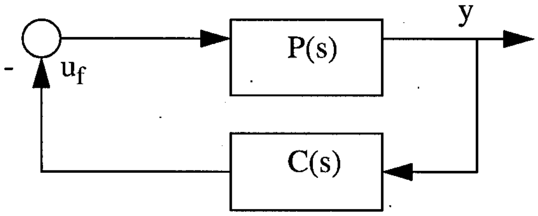


Figure 6.6- Output Feedback Loop

Loop Transfer Recovery is guaranteed if the system between the sensors and the control input is minimum-phase and left-invertible. Also, if there are no infinite zeros, then exact recovery is possible with finite gains. Otherwise, asymptotic recovery is possible (like LQG/LTR). The LTR problem is to minimize the error between $\mathbf{L}_t$(s) and **L**(s), or

$$\boldsymbol{E}(s) = \boldsymbol{L}(s) - \boldsymbol{L}_t(s) \tag{6.27}$$

The error can also be represented in the form of Equation (6.28) [62].

$$E(s) = M_f(s)\,[I_m + M_f(s)]^{-1}\left[I_m + F(sI - A)^{-1}B\right] \qquad (6.28)$$

where $\mathbf{M}_f(s)$ is the recovery matrix, and **F**, **A**, and **B** are determined from the LQR target loop. The $\mathbf{M}_f(s)$ matrix can be further broken down, as shown in Equation (6.29).

$$M_f(s) = F\tilde{M}_f(s) = (sI - A + K_fC)^{-1}(B - K_fD) \qquad (6.29)$$

Thus, the LTR problem reduces down to minimizing the 2-norm of the $\tilde{M}_f(s)$ matrix. The only variable to optimize is the $\mathbf{K}_f$ observer gain matrix.

There are two methods to minimize the $\tilde{M}_f(s)$ matrix - the Riccati approach and the Eigen Assignment approach, both of which are detailed in [63]. For this research, only the Riccati approach was used, so a brief description of this approach is given next. For the Riccati approach, the transpose of $\tilde{M}_f(s)$ is used to represent a fictitious plant to find the optimal $\mathbf{K}_f$. The transpose has the following form:

$$\tilde{M}_f^T = \left(B^T - D^TK_f^T\right)\left(sI - A^T + C^TK_f^T\right)^{-1} \qquad (6.30)$$

The fictitious plant has the form shown in Equation (6.31) and Equation (6.32).

$$\dot{x}_{fict}(t) = A^Tx_{fict}(t) + C^Tu_{fict}(t) \qquad (6.31)$$

$$y_{fict}(t) = B^Tx_{fict}(t) + D^Tu_{fict}(t) \qquad (6.32)$$

With this fictitious plant, the cost function for the 2-norm of $\tilde{M}_f(s)$ reduces down to Equation (6.33).

$$J = \int_0^\infty \left(y_{crit}^{\;T}y_{crit} + \varepsilon u^Tu\right)dt \qquad (6.33)$$

where ε is a very small value. Having ε approach zero is equivalent to having the LQG/LTR process noise go to infinity in the sense that asymptotic recovery is achieved for small values of ε and large values of process noise. The plant is minimized like a normal LQR problem, where we substitute the quadruple (A,B,C,D) by ($\mathbf{A^T}$,$\mathbf{C^T}$,$\mathbf{B^T}$,$\mathbf{D^T}$).

The resulting LTR controller is formed in the same manner as the LQG controller. The only difference is that the LQE optimal estimator gain **K** is replaced by the LTR optimal estimator gain $\mathbf{K}_f$.

6.3.4 SCB/LTR Application

As with the LQG controller, only the non-integral states need estimation. After the $\mathbf{K}_f$ gain is determined, the integral states are reattached as before to find the final controller. The SCB/LTR controller, being a full-order observer design, has the same order as the plant. In this case, the LTR controller is 20th order. We know from the 18th order plant SCB analysis that there are 2 non-minimum-phase zeros, so complete recovery is not possible. Also, there are 2 infinite zeros so at best only asymptotic recovery can be achieved with high gains.

For this design, the full-state feedback gain **F** is simply **Ge**, the LQR gain for the estimated states. Also, since the design must be implemented on hardware, the gains must be limited. For an SCB/LTR design, this is possible by increasing the value of ε placing less emphasis on the minimization of the recovery error. The value used for this design is ε=.01. Table 6.2 shows the estimator $\mathbf{K}_f$ gains achieved for this application.

Table 6.5 - LTR Optimal $\mathbf{K}_f$

Estimated State	y1	y2
x_1	4.0176e-01	-5.0529e+00
x_2	9.2367e-01	-6.7796e+00
x_3	-8.6840e+00	8.6076e+01
x_4	-3.2182e+01	1.0947e+02
x_5	6.8895e+01	9.2625e+01
x_6	1.2994e+01	4.0426e+01
x_7	-1.8468e-01	-2.2199e+01
x_8	2.4941e+00	7.5541e+01
x_9	6.0090e+00	2.7271e+01
x_{10}	-7.6128e+00	5.6034e+01
x_{11}	-5.6783e+01	3.5197e+0
x_{12}	-7.7307e+00	1.1920e+00

Table 6.5 - LTR Optimal K_f

Estimated State	y1	y2
x_{13}	4.6055e-01	6.3234e-01
x_{14}	1.9505e-01	3.1258e+00
x_{15}	8.4164e-02	3.0838e-01
x_{16}	-8.1887e-02	-5.1376e-02
x_{17}	-5.3406e-01	-3.8686e-01
x_{18}	1.6184e-01	3.3696e-01

The controller was implemented at a 1 ms sampling time, like the LQG and LQG/LTR controllers. Simulation results were again identical to LQG/LTR since the simulation assumes perfect initial knowledge of the states. Figure 6.7 shows the experimental evaluation of the SCB/LTR controller on the arm along trajectory 2. Once again, the response is similar to the LQG and LQG/LTR controllers.

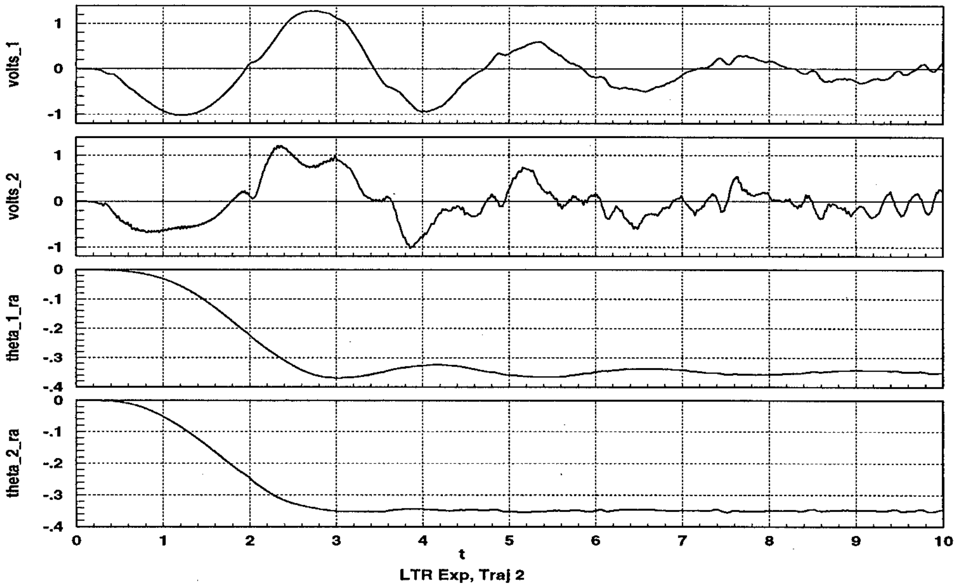


Figure 6.7- LTR Controller Experimental Run - Traj 2

6.4 Controller Reduction

In cases where the plant order is extremely large, often times controller reduction is required for implementation. There are two primary methods for reducing the order of controllers, and both are applied to the LQG/LTR and LTR controllers. It was expected that both controllers would reduce to about the same order since both used the same LQR state feedback gains, which primarily determine the dynamic performance of the system. The first method is balanced truncation [52] and the second is modal residualization [26].

When reducing the controllers, it is important to retain the 2 integral controller states. Thus, these modes are removed, like in the observer design, and added back to the system after reduction of the remaining controller states.

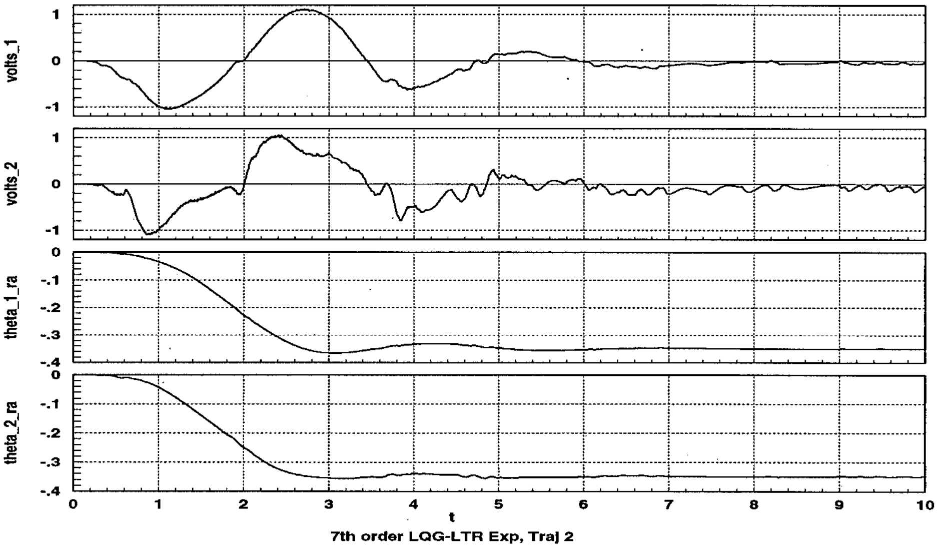


Figure 6.8- LQG/LTR 7th Order Reduced Controller Exp Response- Traj 2

With balanced truncation, the system is balanced and arranged from the largest Hankel singular value (HSV) to the smallest. Since the HSVs represent the controllability and observability of the system, those modes with small HSVs don't contribute much to the dynamic performance of the system. Thus, they can be removed and reduce the system order with minimal loss of performance. When balanced truncation was applied to the LQG/LTR and the LTR controllers, both were only stable for 19th order reduced controllers. The reason for this is that the HSVs were all important (the smallest HSV was about 2 in both cases). Because these controllers had degraded performance in simulation, they were not implemented experimentally.

Modal residualization puts the system in modal form, with all modes decoupled in the controller system matrix. The fastest modes are those with the largest real eigenvalues, and these modes don't have as much of an effect on the performance because they happen so quickly. The slow modes tend to dominate system performance. Application of modal residualization to both controllers produced 7th order controllers with only a small degradation in performance.

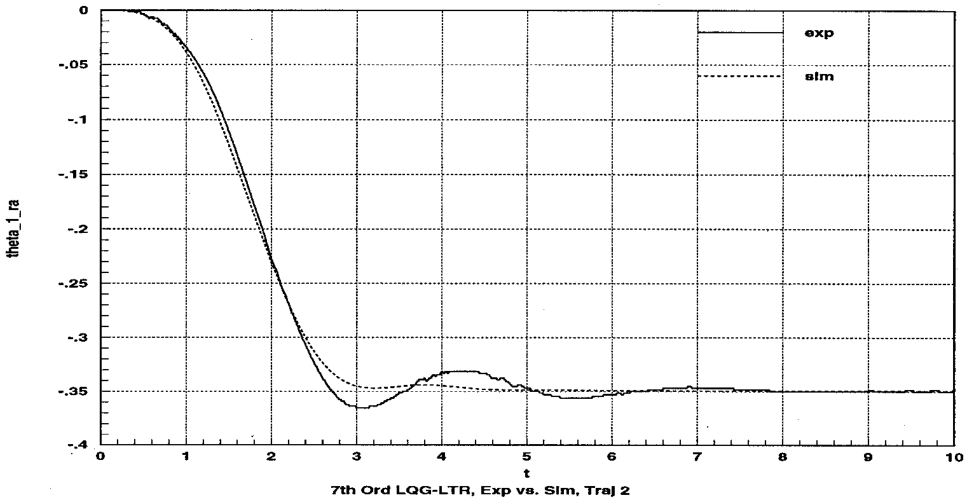


Figure 6.9- LQG/LTR 7th Order Reduced Controller Exp vs. Sim, θ_1 - Traj 2

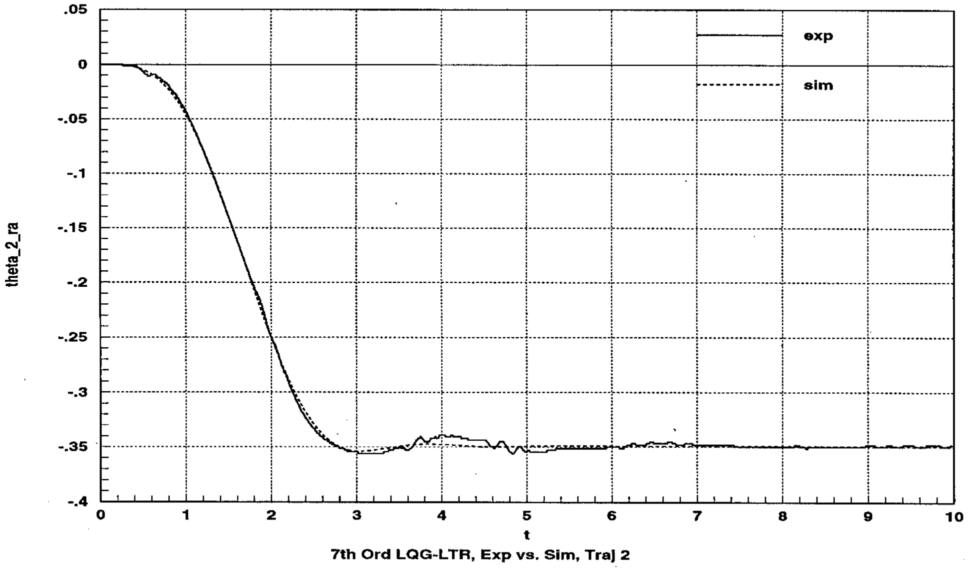


Figure 6.10- LQG/LTR 7th Order Reduced Controller Exp vs. Sim, θ_2- Traj 2

The LQG/LTR 7th order controller was implemented on the arm by applying a Tustin transformation for a 1 ms sampling time. Figure 6.8 shows the experimental response of the 7th order LQG/LTR controller, while Figure 6.9 and Figure 6.10 show the experimental results as compared to simulation for the same controller. Note that the performance is comparable to the full-order LQG/LTR controller and that the simulation catches the correct trends in both links.

The SCB/LTR reduced 7th order controller was also implemented and tested on the arm at 1 ms sampling time. Figure 6.11 displays the experimental response on the arm for trajectory 2, which is similar to the LQG/LTR response. Although not shown, the simulation results are quite similar to the LQG/LTR simulation results.

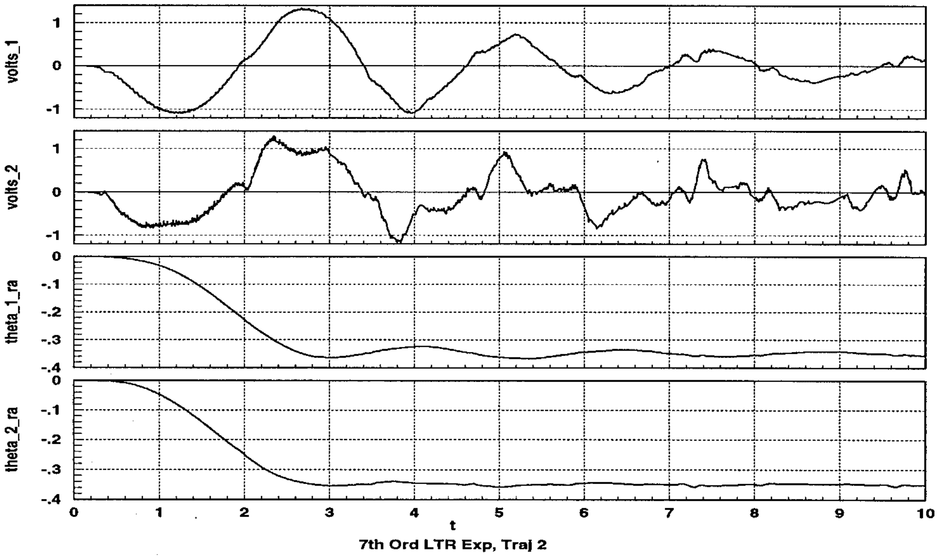


Figure 6.11- SCB/LTR 7th Order Reduced Controller Exp Response- Traj 2

6.5 Control Bandwidths

Since all the controllers are based on the same LQR gains, and the performance of each are similar, they should have similar control bandwidths. It was noticed that the LQG-class controllers never experienced high-frequency vibrations, as seen from the evaluation of the control bandwidths. Figure 6.12 shows the θ_1/θ_{1cmd} control bandwidth plot, while Figure 6.9 shows the θ_2/θ_{2cmd} control bandwidth plot. The bandwidth for the first loop is about 2 rad/s, while the second loop also has a bandwidth of about 2 rad/s.

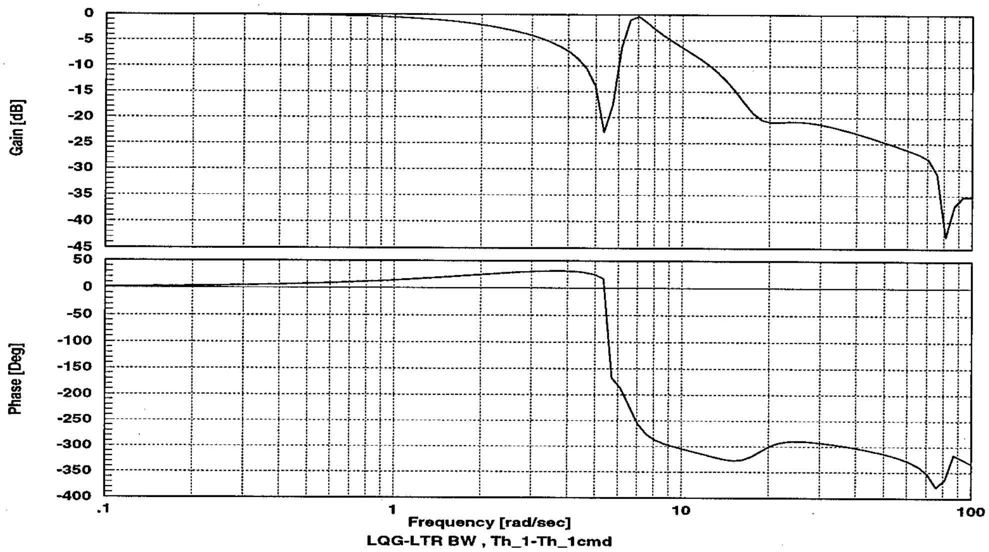


Figure 6.12- θ_1/θ_{1cmd} LQG Control Bandwidth

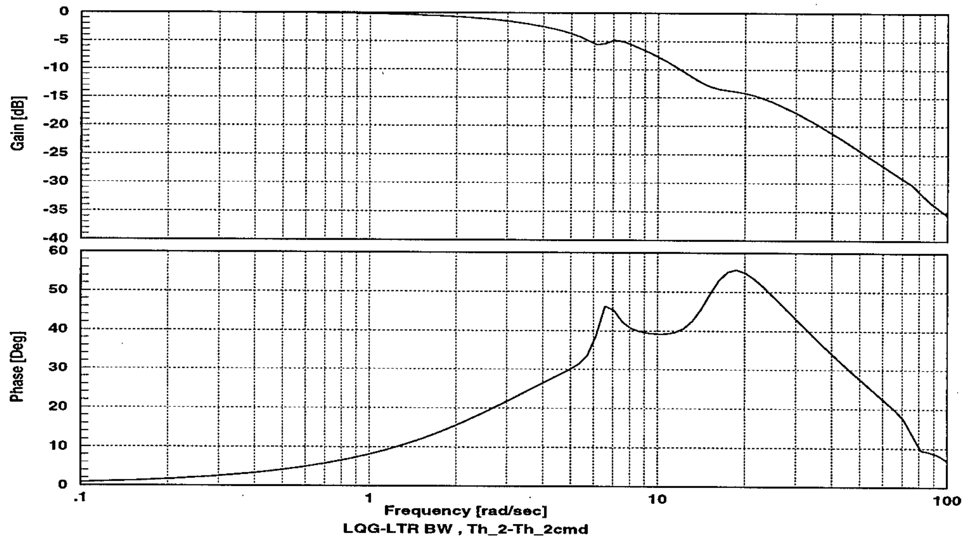


Figure 6.13- θ_2/θ_{2cmd} LQG Control Bandwidth

6.6 Observations

Clearly, the reduced-order controllers provide comparable performance to the full-order controllers. Thus, implementation of LQG-class controllers may allow the use of reduced-order controllers. It is interesting to note that the 20th order controllers ran at a 0.4 ms sampling time, while the 7th order controllers ran at a 0.3 ms sampling time. In comparison, 4th order controllers and PID controllers ran at 0.2 ms sampling times. The fastest sampling time allowed by the data acquisition system is 0.1 ms. Thus, by reducing the controller order enough, you can sample at twice the rate. This might be critical in some applications.

Appendix C shows additional plots for all of the LQG-class controllers for different trajectories. In addition, comparison plots show how each controller performs along the different trajectories. Performance of the controllers remain about the same regardless of the trajectory.

7 SANDY Background

Many MIMO designs make use of optimal control through linear-quadratic-regulator (LQR) and linear-quadratic-gaussian (LQG) techniques. Both LQR and LQG are based on solutions to the appropriate Riccati equations, and are well-documented [1] [35]. LQR has the advantage of guaranteed robustness but the disadvantage of requiring full-state feedback. LQG has the advantage that it can be implemented based on available sensors, and the disadvantage of a loss in robustness and the use of high controller order (the same order as the plant). To remedy the loss of robustness, LQG/LTR Loop Transfer Recovery (LTR) has been developed [20]. However LQG/LTR still suffers the disadvantage of a high-order controller.

To combat the high controller order, techniques such as balanced truncation or modal residualization are applied to the controller in hopes that a satisfactory reduced order controller can be obtained. Unfortunately, there is no guarantee that a suitable low-order controller can be found through model reduction. A systematic approach to the design of low-order controllers has been developed by Ly and his coworkers [44] [45] [46]. This approach is based on direct optimization and is implemented in the computer code SANDY [45]. SANDY allows specification of additional design constraints and of the desired controller structure.

In order to use SANDY, plant models must be represented in the format described in equations (7.1)-(7.3). Figure 7.1 gives a block diagram view of this plant description, which has a typical feedback control system structure.

The plant models are described by

$$\dot{x}^i(t) = F^i x^i(t) + G^i u^i(t) + \Gamma^i w^i(t) \tag{7.1}$$

$$y_c^i(t) = H_c^i x^i(t) + D_{cu}^i u^i(t) + D_{cw}^i w^i(t) \tag{7.2}$$

$$y_s^i(t) = H_s^i x^i(t) + D_{su}^i u^i(t) + D_{sw}^i w^i(t) \tag{7.3}$$

where

- $x^i(t)$ is a state vector of dimension n.
- $u^i(t)$ is a control vector of dimension of m.

- $w^i(t)$ is a disturbance/command input vector of dimension m'.
- $y_s^i(t)$ is a measurement vector of dimension p.
- $y_c^i(t)$ is a vector of dimension p' containing the design criterion variables.

The superscript i corresponds to the ith plant in a multiple plant setting.

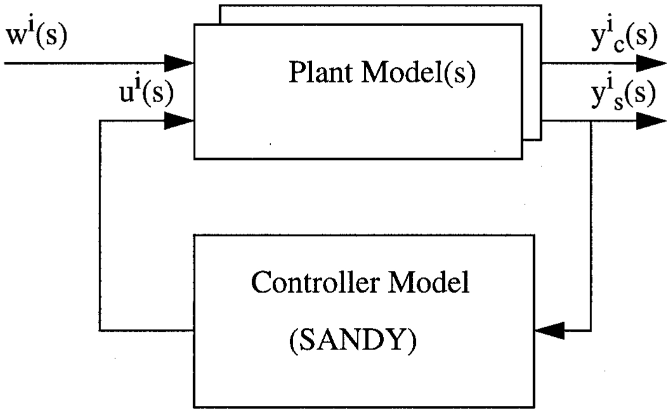


Figure 7.1- Sandy Feedback Control Structure

The SANDY program optimizes the following design objective function given by

$$J(t_f) = \sum_{i=1}^{N_{pm}} w_{pi} J_i\left(t_f^i\right) \tag{7.4}$$

where the term $J_i(t_f^i)$ ($i=1,N_{pm}$) is a quadratic performance index evaluated at the ith plant, and N_{pm} is the number of design plant models. Also, each plant may have a weighting denoted by w_{pi}. Notice that each term has a variable final time t_f^i to allow evaluation of the design objective for systems which are initially unstable.

For the deterministic case (initial condition response), the cost function is:

$$J_i\left(t_f^i\right) = (1/2)\int_0^{t_f^i} e^{2\alpha_i t}\left[y_c^{iT}(t)\,Q^i y_c^i(t) + u^{iT}(t)\,R^i u^i(t)\right]dt \tag{7.5}$$

For random impulsive disturbances, the objective function is:

$$J_i\left(t_f^i\right) = (1/2)\int_0^{t_f^i} e^{2\alpha_i t} E\left[y_c^{iT}(t)\,Q^i y_c^i(t) + u^{iT}(t)\,R^i u^i(t)\right]dt \qquad (7.6)$$

where the expectation operator E[] is used to handle random inputs. Finally, for random white-noise disturbances, the following equation is used:

$$J_i\left(t_f^i\right) = E\left[y_c^{iT}\left(t_f^i\right)Q^i y_c^i\left(t_f^i\right) + u^{iT}\left(t_f^i\right)R^i u^i\left(t_f^i\right)\right] \qquad (7.7)$$

In all the above cost functions, the Q^i terms are weighting on the criterion outputs for each plant, and R^i is the weighting on the controls. The $e^{\alpha_i t}$ term is a destabilization term for each plant.

In addition, constraints can be applied to the optimization. Some of the constraints available in SANDY are direct bounds on the design parameters in the SANDY controller matrices, nonlinear constraints on the covariance responses of the outputs and controls, and nonlinear constraints on the closed-loop stability by specifying the maximum real part of the closed-loop eigenvalues and the minimum damping of the complex roots. Additionally, the command inputs can be filtered to match the desired input responses.

Once all the constraints and the weightings have been specified, the designer then chooses the structure of the controller by specifying an initial guess to the controller A,B,C,D matrices. Next, the variables to be optimized are specified. SANDY then minimizes the cost function of equation (7.4) with respect to the design parameters in the A,B,C,D matrices of the controller and comes up with a single controller for all the design plant conditions.

8 SANDY Double-Angle Control Design

The simplest design is to optimize SANDY to the error between the reference commands and the actual joint angles. Thus, the optimization is based on the two measured angle sensors and the two angle reference trajectories.

8.1 SANDY Design

For this design problem, one plant condition was developed with a filtered step command input for the first joint angle, and a second plant with a filtered step command input for the second joint angle. The criterion variables are $\dot{\theta}_1$, $\dot{\theta}_2$, error_θ_1, and error_θ_2, respectively. There are two controls, torque_1 and torque_2 and two sensors, θ_1 and θ_2. The criterion variables are used only in the cost function, and can contain variables other than those available as sensors. Thus, the SANDY controller should have two inputs (the sensors) and two outputs (the torques). Values of the nonlinear constraints were a direct constraint on the control covariance of torque 1 for plant 1 of 10 Nm, a constraint on the maximum real part for the closed-loop eigenvalue of -.25, no weighting on the control penalties R_1 and R_2, and Q_1=diagonal(0,0,1,0) with Q_2 = diagonal(0,0,0,1). The initial guess of the SANDY controller, based on the PID controllers in the earlier chapters, is:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -8 & 0 \\ 0 & 0 & 0 & -10 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 0.1 & 0 & 8 & 0 \\ 0 & 0.11 & 0 & 10 \end{bmatrix}, D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (8.1)$$

This is a fourth order controller with 2 integral states and a first order filter for each of the θ_1 and θ_2 sensor inputs. The integral states are added to reduce the steady state error in the tracking of the joint angles. The controller design variables are A(3,3), A(4,4), B(3,1), B(4,2), C(1,3), and C(2,4) for the first order filters. In addition, C(1,1),C(1,2),C(2,1), and C(2,2) are the design variables for the integral gains. Finally, the entire D matrix is optimized for the proportional gains.

After 300 function calls, the design optimization produced the following SANDY controller

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -3.91 & 0 \\ 0 & 0 & 0 & -7.37 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -7.80 & 0 \\ 0 & -14.93 \end{bmatrix} \tag{8.2}$$

$$C = \begin{bmatrix} 0.172 & -3.31 & 11.08 & 2.97 \\ 0.024 & 3.60 & -2.91 & 11.0 \end{bmatrix}, D = \begin{bmatrix} 25.01 & -1.20 \\ -7.27 & 29.11 \end{bmatrix}$$

The cost function value associated with plant 1 is 0.1345 and the cost for plant 2 is 0.0311.

The controller is then discretized using a Tustin transformation at a 1 ms sampling time. The difference between a 1 ms simulation and a continuous simulation was negligible for this controller, verifying the initial assumption that a continuous SANDY design should provide a good digital controller at this low sample time (a 1000 Hz sampling rate). If digital considerations are desired, more stringent constraints may be placed on the control covariances to keep the gain on the controller low making it more applicable for higher sampling times (and thus lower sampling rates).

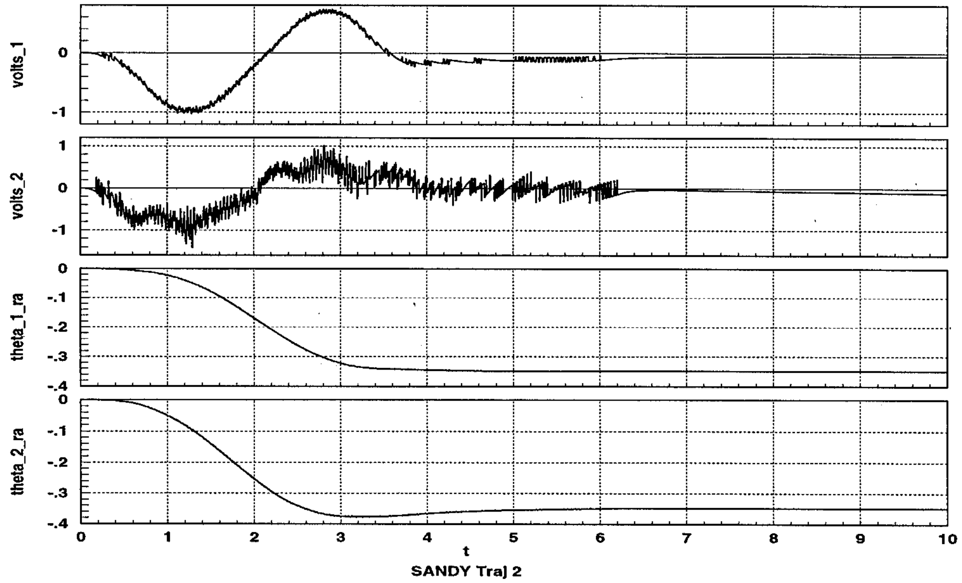


Figure 8.1- SANDY Experimental Run, Trajectory 2

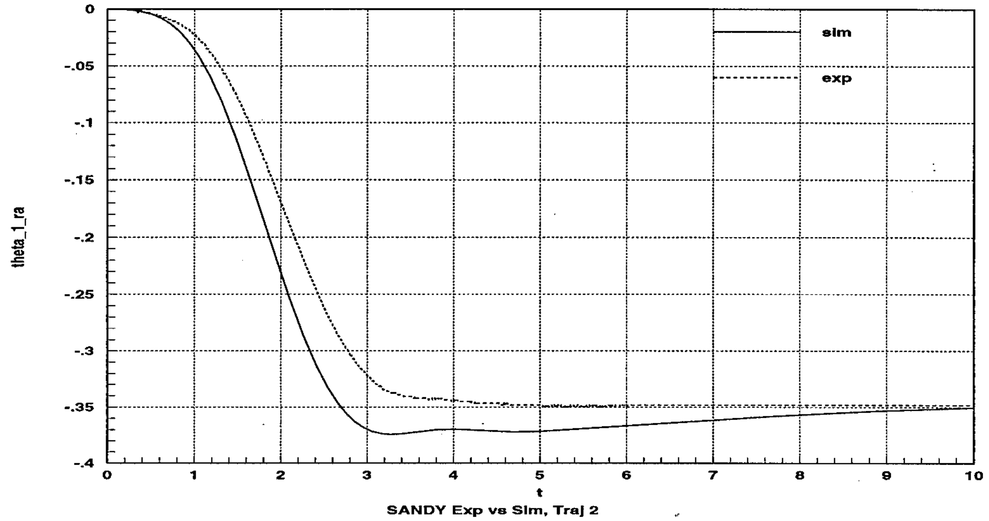


Figure 8.2- SANDY Experimental vs. Simulation, Trajectory 2, θ_1

Figure 8.1 shows the experimental run along trajectory 2. Figure 8.2 and Figure 8.3 show the experimental results vs. simulation for the SANDY controller. Notice that the simulation captures much of the actual dynamics of the experiment. The SANDY controller responds quickly with minimal overshoot and small steady-state error.

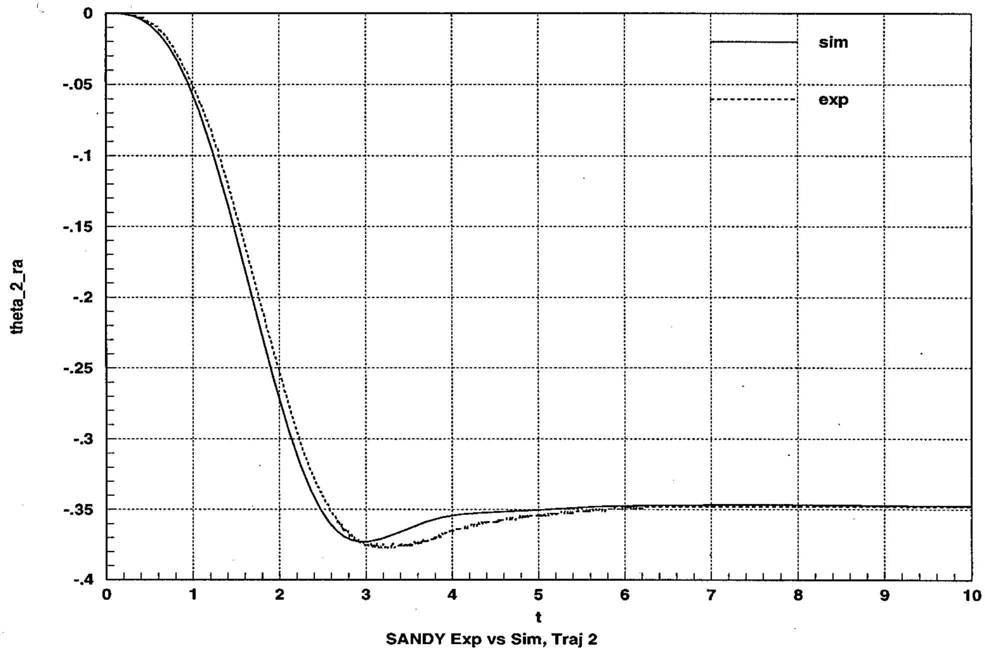


Figure 8.3- SANDY Experimental vs. Simulation, Trajectory 2, θ_2

8.2 SANDY Control Bandwidth

It is important to realize the capability of the controller when evaluating the experimental performance. For the TLA, the control bandwidth is defined from the Bode plot of θ_1/θ_{1cmd} for the first motor and θ_2/θ_{2cmd} for the second motor. Again, the convention used in this research is that the bandwidth is where the magnitude first crosses minus 3dB magnitude. If the control task is greater than the bandwidth, then the controller won't be able to achieve the desired performance. The primary control bandwidth for the first motor (shoulder) is about 3.5 rad/sec as shown in Figure 8.4, while the primary control bandwidth for the second motor (elbow) is around 11 rad/sec as shown in Figure 8.5. Notice that in both cases, there are multiple 0 dB crossings due to high frequency resonances associated with the flexible modes.

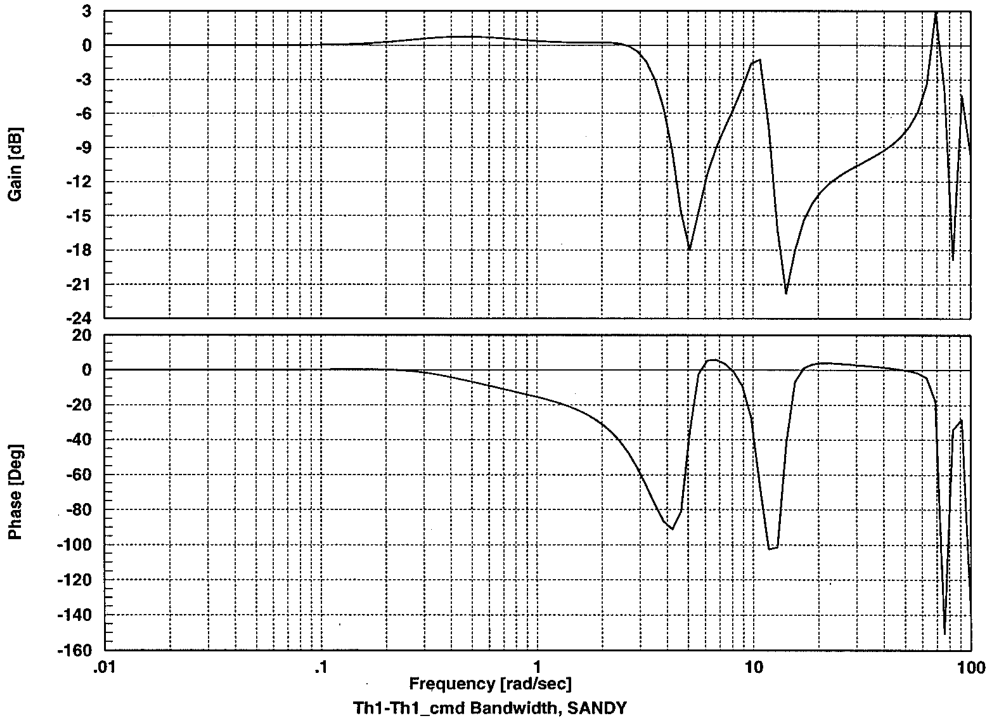


Figure 8.4- Control Bandwidth for the Shoulder Motor

Notice that both bandwidth plots have peaks above 0 dB, meaning those frequencies would be amplified. However, in actuality the commands are filtered through a

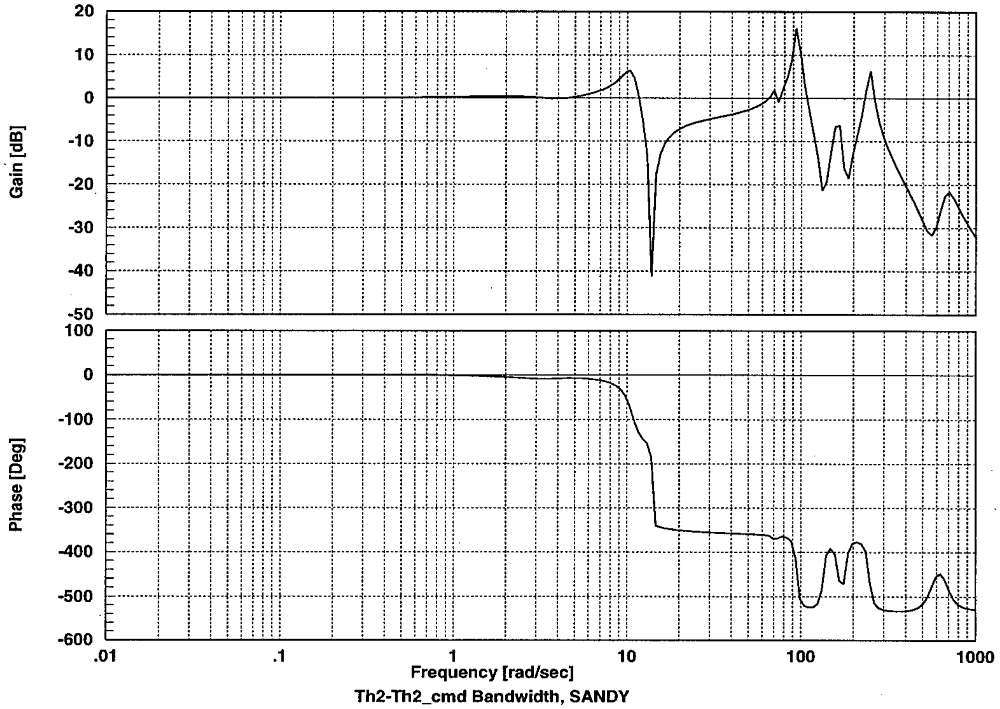


Figure 8.5- Control Bandwidth for the Elbow Motor

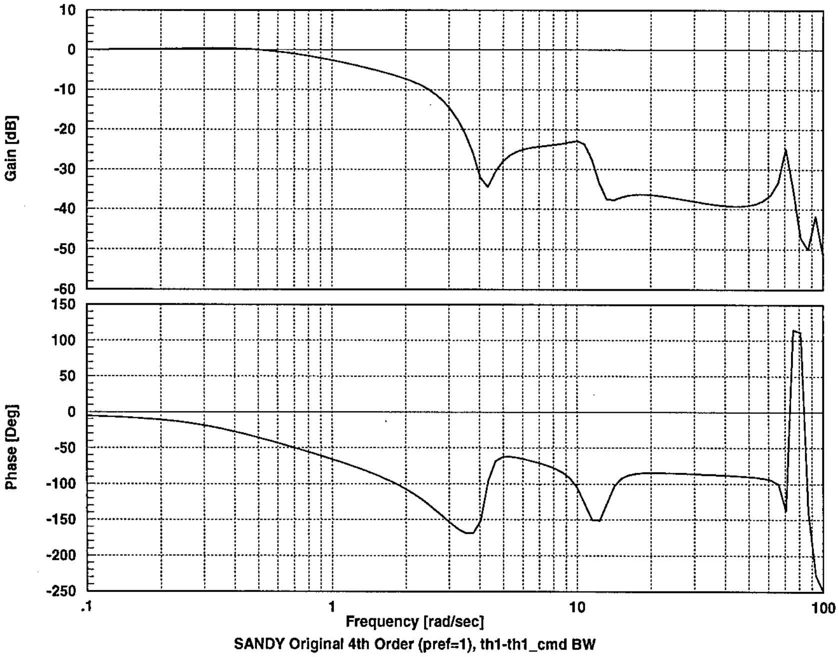


Figure 8.6- SANDY Bandwidth With Prefilter, Shoulder Motor

spline, and this must be accounted for in the bandwidth plots. Also, the SANDY optimization assumes a 1st order command prefilter at 1 rad/s is present. Thus, bandwidth plots with the command prefilter are included. Figure 8.6 shows the filtered bandwidth for the shoulder motor is about 1 rad/s while Figure 8.7shows that the elbow motor also has a control bandwidth of 1 rad/s. Notice that with the command prefilter at 1 rad/s the magnitude never exceeds 0 dB, which is desirable for a control bandwidth response.

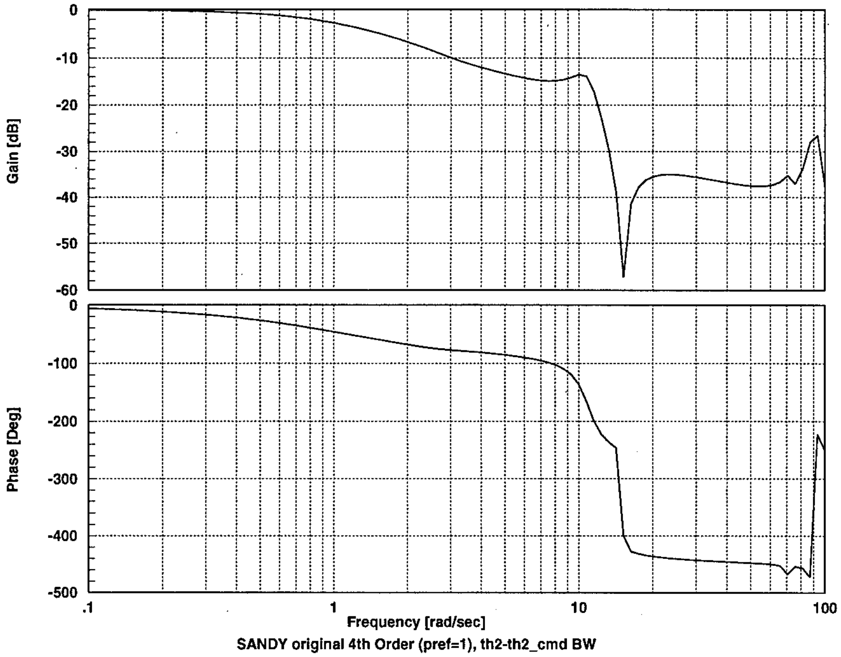


Figure 8.7- SANDY Bandwidth With Prefilter, Elbow Motor

8.3 Experimental Comparison of SANDY and PID Controllers

The purpose of this section is to compare the experimental results of the SANDY and PID controllers. Although in this case the SANDY controller displays better overall response in terms of tracking error, it should be pointed out that more design time was spent on the SANDY controller than on the PID controller. Since both are effectively 4th order controllers, they should have similar performance if an equal amount of effort is spent on each of the designs. The one advantage the SANDY controller has over a straight PID, however, is the ability to provide crossfeed information from one angle into the control for the other. This is up to the designer in the choice of the SANDY structure.

The baseline comparison is for all the three trajectories listed in Chapter 5. Trajectory 2 comparisons are found in Figure 8.8 and Figure 8.9.

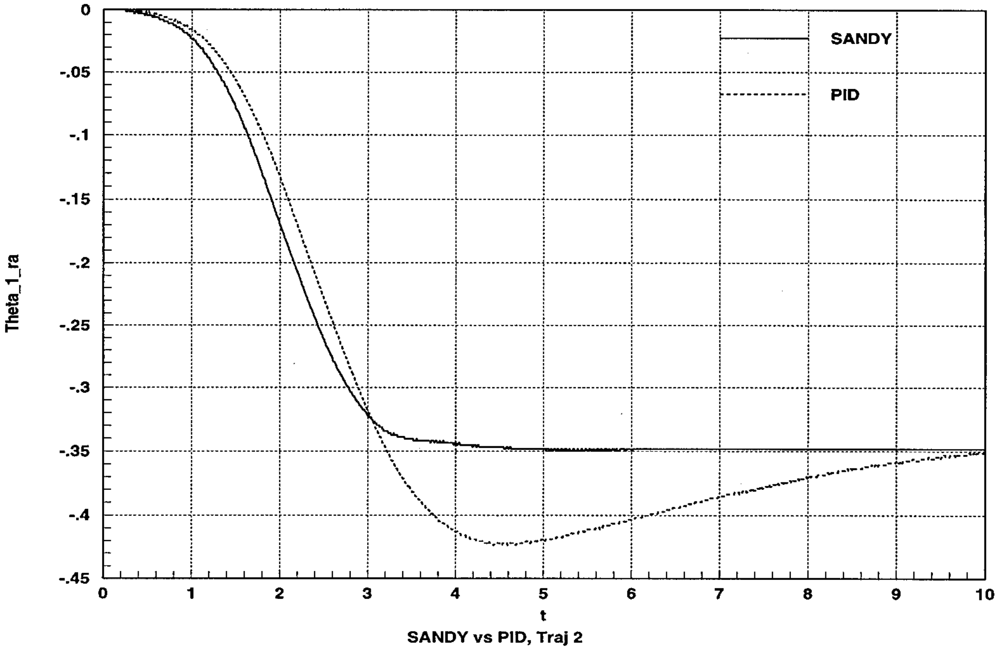


Figure 8.8- PID vs. SANDY, Traj 2, θ_1

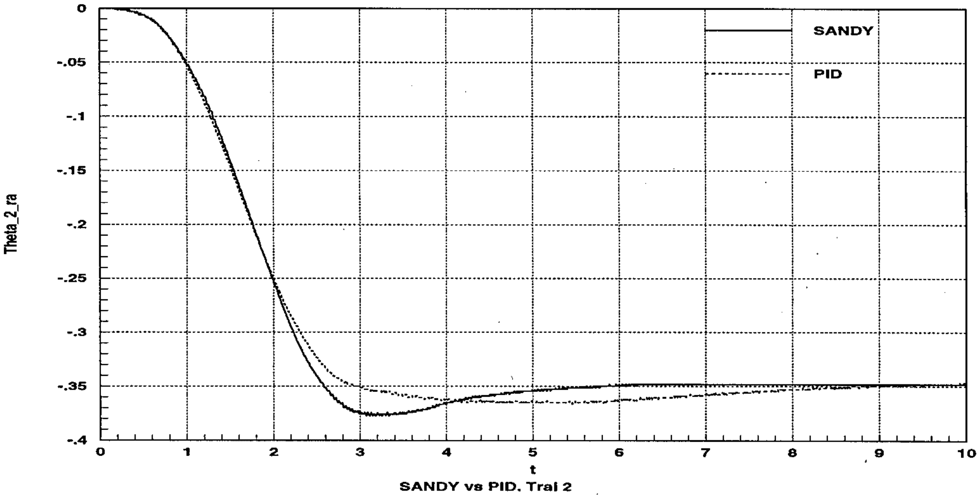


Figure 8.9- PID vs. SANDY, Traj 2, θ_2

The SANDY controller responds more rapidly and with less overshoot than the PID controller for link 1 and has comparable overshoot but with lower settling time in link 2.

To test the robustness of each respective controller, a different tip attachment is used. The new tip, bracket, and bowl weigh 725 grams total compared to 351 grams for the original tip.

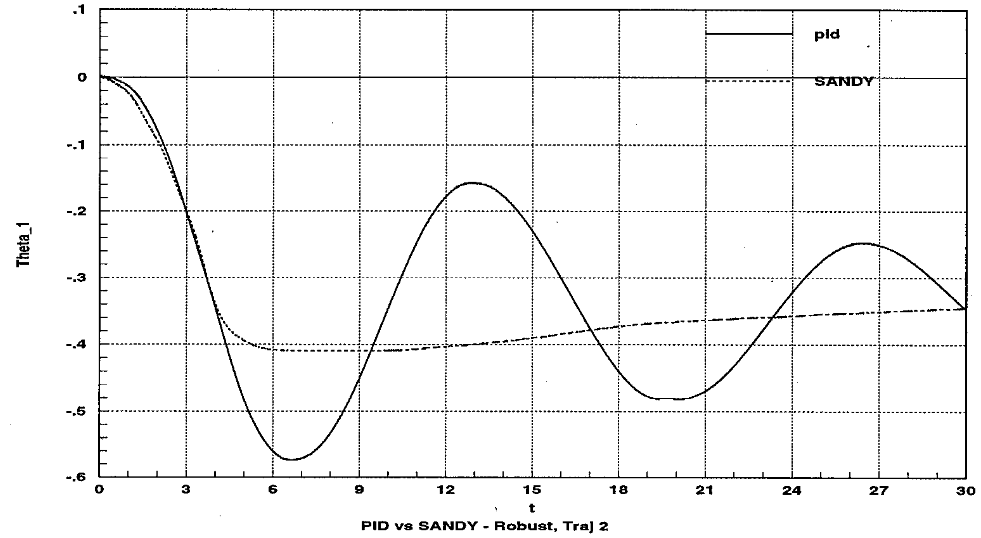


Figure 8.10- Robust PID vs. SANDY, Traj 1, θ_1

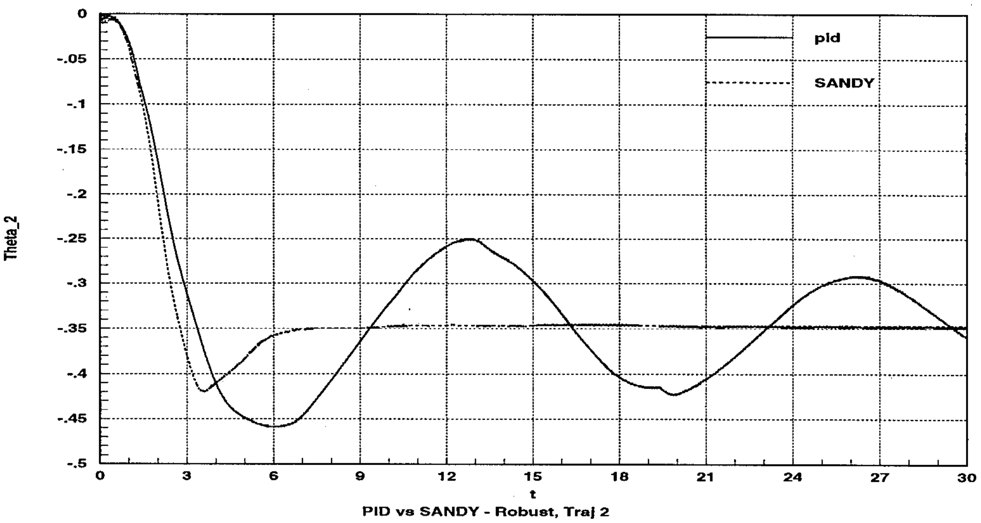


Figure 8.11- Robust PID vs. SANDY, Traj 1, θ_2

In addition, to provide variable dynamics, a heavy metal ball with a weight of 1.043 kg is placed in the middle of the cup and is free to move throughout the slew. Trajectory 2 results are displayed in Figure 8.10 and Figure 8.11. This time, the SANDY controller response was much faster and had significantly less overshoot.

Appendix B contains additional plots of the SANDY controller along different trajectories. In addition, comparison plots with the PID controller are included along the different trajectories. The trends noted for trajectory 2 in this chapter are the same for all the tested trajectories in Appendix B.

9 Sandy Robust Design and Endpoint Position Test

As an alternative to controlling the hub endpoint position through the hub angles, the optimization can be reformulated using forward and inverse kinematics in the model to control the x and y endpoint positions.

9.1 Forward Kinematics

The endpoint position can be predicted from the hub angles through the forward kinematics. It is important to realize that this technique is very good for rigid manipulators, but provides only an estimate of the endpoint position for flexible manipulators. Figure 9.1 shows the conventions for forward kinematics.

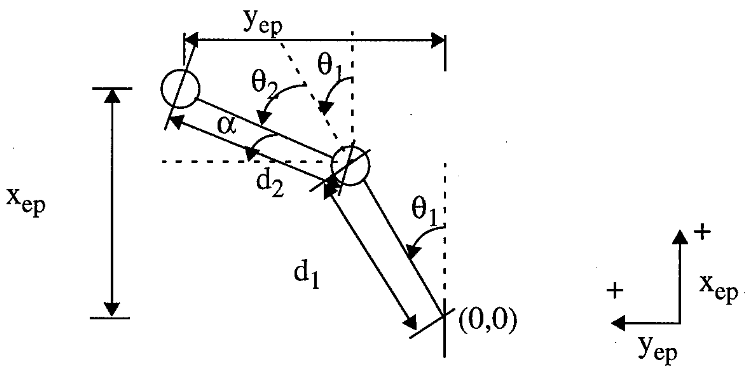


Figure 9.1- Forward Kinematic Conventions

The final endpoint position is actually a contribution from the link 1 and link 2 positions. The length of the links, d_1 and d_2, is related to parameters listed in Appendix A.

$$d_i = o_{hi} + L_i + o_{ti} \;;i= 1,2 \tag{9.1}$$

If the desired endpoint is at the end of the end-effector, then o_{t2} can be increased to move the endpoint to the desired location. For the arm used in this research, d_1=0.6363 m and d_2 = 0.6096 m.

The joint angle θ_1 is associated with the first link and the angle θ_2 corresponds to the second link. To obtain the endpoint position, a new angle α is defined in Equation (9.2).

$$\alpha = 90 - \theta_1 - \theta_2 \tag{9.2}$$

Using these equations and definitions results in the following equations for x_{ep} and y_{ep}.

$$x_{ep} = d_1 \cos(\theta_1) + d_2 \sin(\alpha) \tag{9.3}$$

$$y_{ep} = d_1 \sin(\theta_1) + d_2 \cos(\alpha) \tag{9.4}$$

The sign convention for this scheme is CCW = + for the angles and + is based on the right-hand rule for the cartesian directions, as shown in Figure 9.1.

9.2 Inverse Kinematics

The inverse kinematics take the known x_{ep} and y_{ep} locations, and calculate the θ_1 and θ_2 angles. This is trickier than the forward kinematic case because there are an infinite number of mathematical solutions for θ_1 and θ_2 given a specific x_{ep} and y_{ep}. Fortunately, there are only 2 physical redundancies for the angle positions, as shown in Figure 9.2. The rest are just mathematical equivalents for the 2 physical angles, differing by adding multiples of 360°.

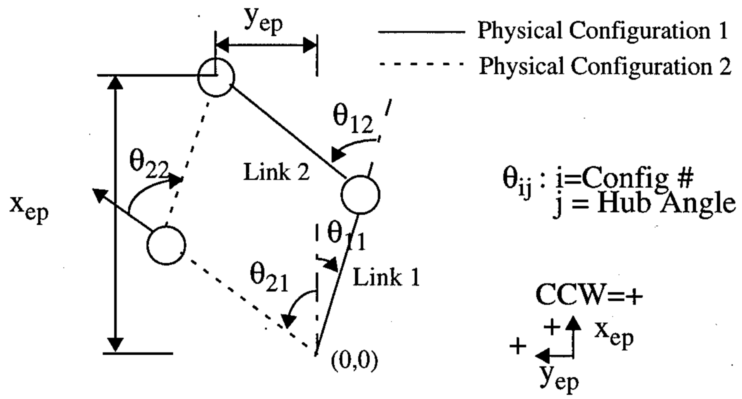


Figure 9.2- TLA Physical Angle Redundancy

For the TLA manipulator used in this research, only physical configuration 1 is possible. The second motor has stops which prevent θ_2 from ever reaching negative values, so our physical system has only one possible inverse kinematic solution. Making use of the physical configuration makes a unique analytic solution possible by ensuring the θ_2 angle is always positive. The inverse kinematics can be solved using the law of cosines for the triangle shown in Figure 9.3.

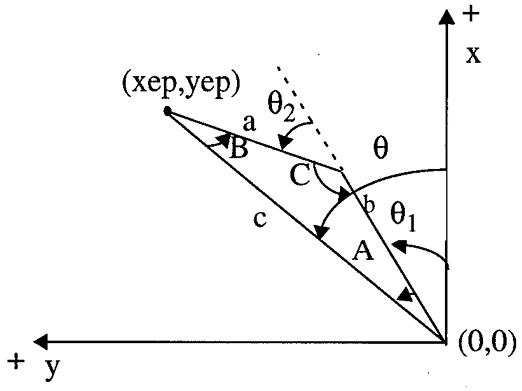


Figure 9.3- Inverse Kinematic Angle Conventions

The general form of the law of cosines is

$$a^2 = b^2 + c^2 - 2bc\cos(A) \tag{9.5}$$

Table 9.1 lists the correspondence between the generic terms in the law of cosines and the TLA application.

Table 9.1 - Law of Cosine TLA Terms

Generic Term	TLA Term	Units
a	d_2	m
b	d_1	m
c	r	m
A	θ-θ_1	rad
B	N/A	rad
C	180-θ_2	rad

where r and θ are the polar coordinates for the endpoint. Specifically,

$$r = \sqrt{\left(x_{ep}^2 + y_{ep}^2\right)} \tag{9.6}$$

and

$$\theta = \operatorname{atan}\left(\frac{y_{ep}}{x_{ep}}\right) \tag{9.7}$$

Solving for angle A using the law of cosines and substituting the values from Table 9.1 yields and analytic expression for θ_1.

$$\theta_1 = \theta - \operatorname{acos}\left(\frac{\left(d_1^2 + r^2 - d_2^2\right)}{2d_1 r}\right) \tag{9.8}$$

Similarly, solving for angle C yields an expression for θ_2.

$$\theta_2 = 180 - \operatorname{acos}\left(\frac{\left(d_1^2 + d_2^2 - r^2\right)}{2d_1 d_2}\right) \tag{9.9}$$

9.3 TLA Application

The optical encoder mounted to the second motor shaft is oriented so that its index or zero reading point is actually at 90°. Thus, to achieve the correct kinematics, 90° must be added to the angle read from the encoder for θ_2. The resulting kinematic relationship is shown in Table 9.2.

Table 9.2 - Some TLA Kinematic Relationships

θ_1 (°)	θ_2 (°)	θ_2+90 (°)	x_{ep} (m)	y_{ep} (m)
0	0	90	.6363	.6096
0	-90	0	1.246	0
0	-20	70	.8448	.5728
0	+20	110	.4278	.5728
-20	0	90	.8064	.3552
-20	-20	70	.9898	.2494
+20	0	90	.3894	.7905

Table 9.2 - Some TLA Kinematic Relationships

θ_1 (°)	θ_2 (°)	θ_2+90 (°)	x_{ep} (m)	y_{ep} (m)
+20	+20	110	.2061	.6846
-10	-10	80	.8351	.4623
+10	+10	100	.4181	.6833
-40	-40	50	1.09	-.300

It is interesting to note that when dealing with the endpoint position, the actuators are still located at the hubs. In other words, the endpoint position must be converted to hub angles through inverse kinematics to produce angle reference signals. The actual angles read from the encoders are compared to the reference signals to produce error signals, which are fed into the joint controller. Figure 9.4 shows the block diagram implementation for such a control scheme.

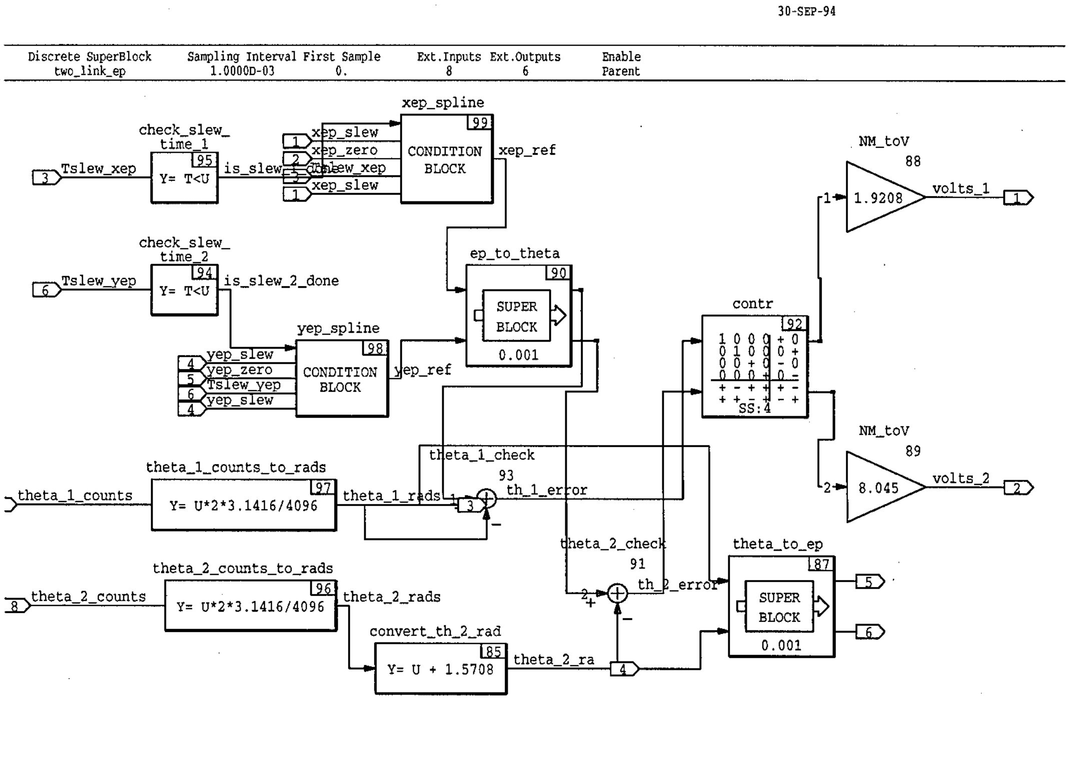


Figure 9.4- TLA Endpoint Trajectory Implementation

9.4 Endpoint Trajectory Generation

In order to test endpoint tracking, several trajectories are used. These trajectories provide differing dynamic situations for the TLA, and provide a test of robustness to trajectories.

9.4.1 Straight Line

One of the simplest yet most common trajectories in robotics is a straight line. Going between any 2 points in two-dimensional cartesian space constitutes a straight line [56]. In order to smooth the commands, a quintic spline in x and y is commanded between the start and end points. This is similar to the quintic spline used to command hub angles. The resulting reference trajectory is shown in Figure 9.5.

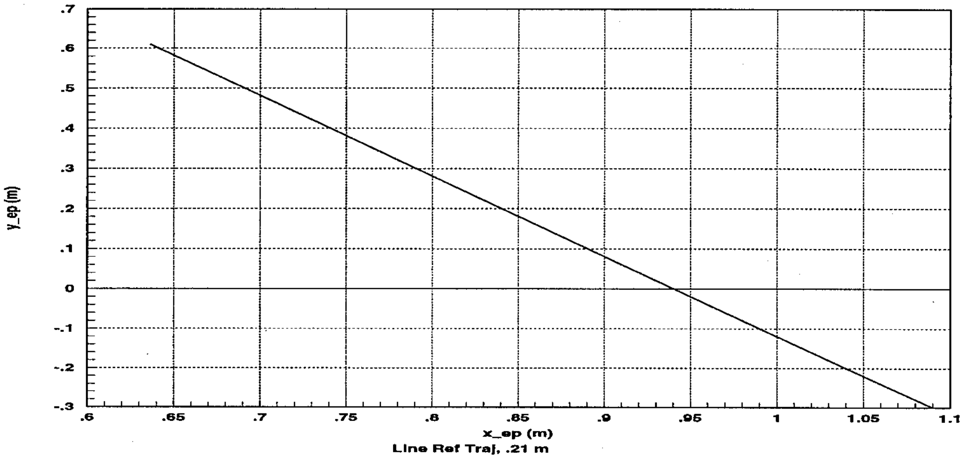


Figure 9.5- 0.21m Straight Line Reference Trajectory

It is interesting to compare the hub angles and endpoint positions for a spline input of the angles versus a spline input of the endpoint. Figure 9.6 shows the reference profiles given hub spline inputs while Figure 9.7 shows the reference given endpoint spline inputs.

Notice in Figure 9.6 that while the hub angle commands are the same for a simultaneous slew of -40^{o} in 3 seconds, the x_{ep} and y_{ep} references are actually different. In this case, the x_{ep} reference reaches the desired position in about 2 seconds while the y_{ep} reference takes about 3 seconds. Interestingly, both x_{ep} and y_{ep} approach their final positions in a spline like manner for this trajectory.

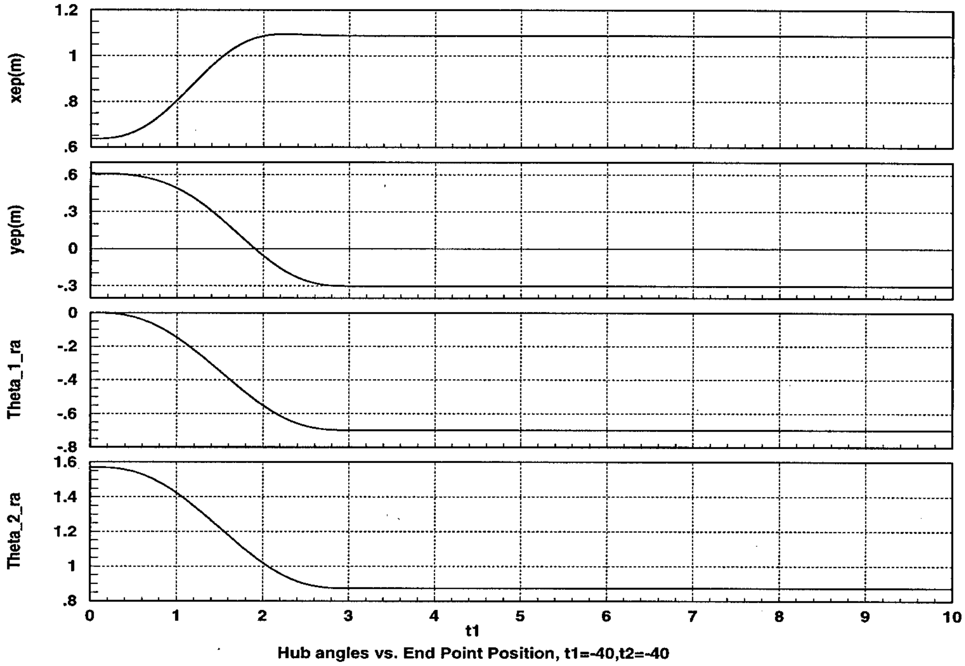


Figure 9.6- Hub Angle Spline Input Reference Trajectory

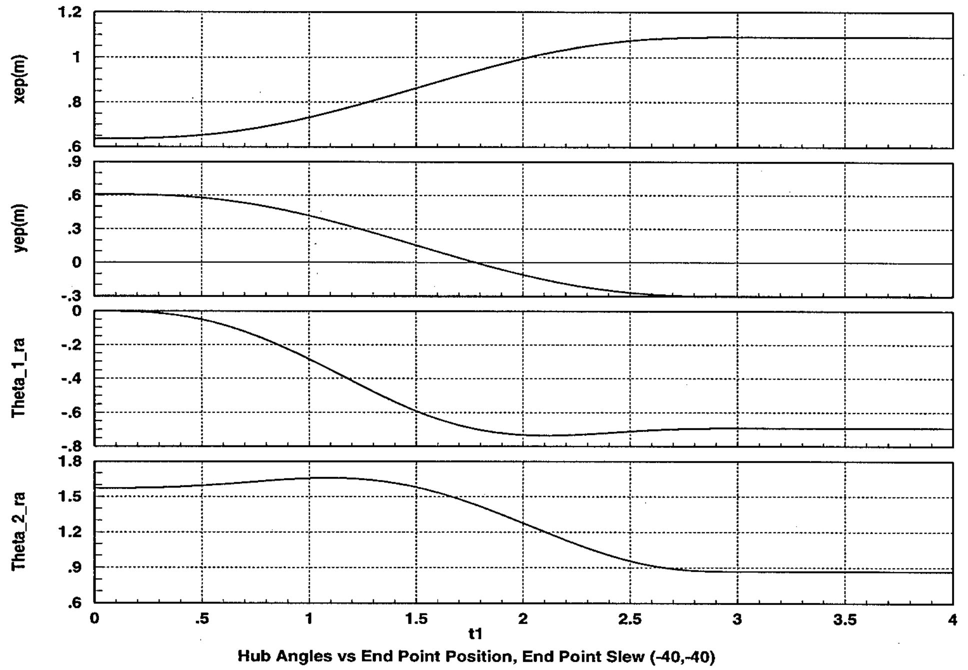


Figure 9.7- Endpoint Spline Reference Trajectory

When commanding the same slew through endpoint positions, slightly different hub reference commands are generated, as shown in Figure 9.7. Now, the hub commands have slight overshoot and undershoot in the references and are not ideal spline inputs. Also, the θ_1 reference reaches the required end angle in about 2 seconds while the θ_2 reference reached its final value in 3 seconds.

9.4.2 Arc

Another common trajectory which tests the dynamics of a controller is a simple circular arc. By specifying the start point, radius of curvature, arc length and slew time, many different variations can be tested. As a start, an arc similar to a physical arc used in trajectory tracking is tested. The arc has a radius of curvature of 0.51m, starts at the θ_1=0° and θ_2=90°, and travels 40°. Figure 9.8 shows the reference trajectory for this arc in terms of x_{ep} and y_{ep}.

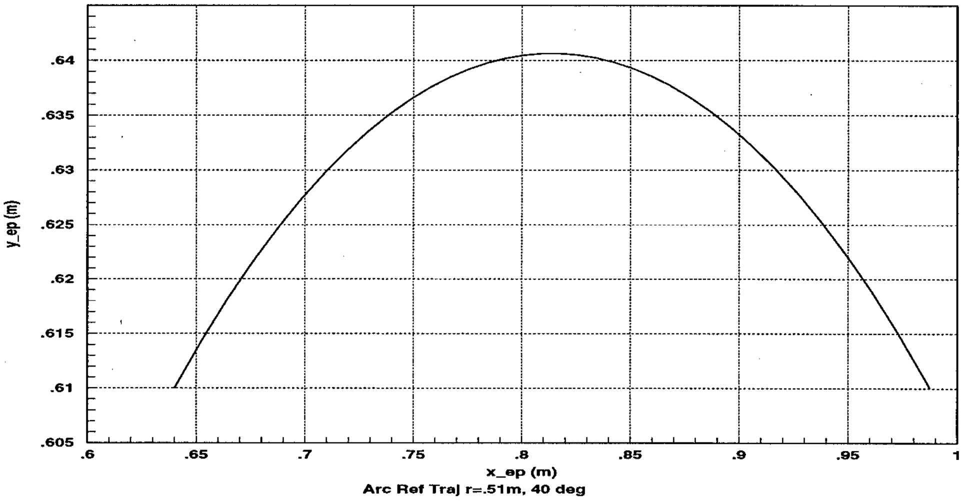


Figure 9.8- 40° Arc Reference Trajectory

This arc is implemented mathematically using the following algorithm. First, given the starting point, radius, and arc length, the coordinates of the center of the circle corresponding to the arc are calculated. Then, the angle increment based on percentage of slew time is calculated to give an angle position relative to the start of the arc. Knowing the angle position and center of the circle, the current endpoint position on the arc is calculated. When the time is greater than the desired slew time, the endpoint of the arc is the reference. Figure 9.9 shows a graphical depiction of the algorithm.

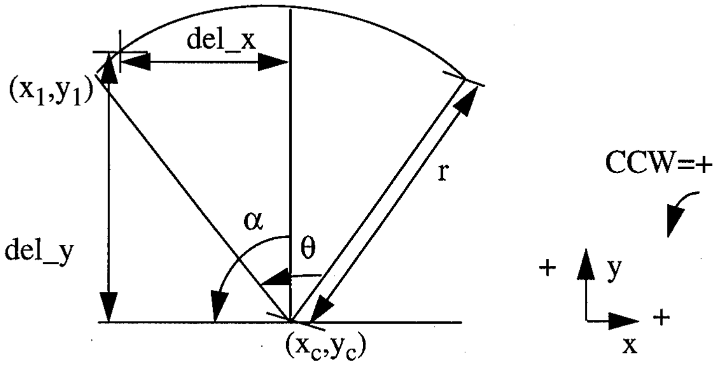


Figure 9.9- Graphical Depiction of the Arc Trajectory

Mathematically, the algorithm is defined by the following equations.

$$\Delta x_1 = r\sin\left(\frac{\pi}{2}-\frac{\theta}{2}\right) \tag{9.10}$$

$$\Delta y_1 = r\cos\left(\frac{\pi}{2}-\frac{\theta}{2}\right) \tag{9.11}$$

Since the starting point (x_1,y_1) is known, the center point can be found.

$$x_c = x_1 + \Delta x \tag{9.12}$$

$$y_c = y_1 - \Delta y \tag{9.13}$$

Once the center point is known, the trajectory is calculated based on the desired slew time.

$$\Delta\theta = \frac{\theta}{\frac{(Tslew)}{t}} \tag{9.14}$$

By knowing the change in the arc angle, the change in Δx and Δy can be calculated from the following equation.

$$\Delta x = r\cos\left(\frac{\pi}{2}-\frac{\theta}{2}-\Delta\theta\right) \tag{9.15}$$

$$\Delta y = r\sin\left(\frac{\pi}{2}-\frac{\theta}{2}-\Delta\theta\right) \tag{9.16}$$

Finally, the x and y arc locations along the arc are then known.

$$x = x_c - \Delta x \quad (9.17)$$

$$y = y_c + \Delta y \quad (9.18)$$

When the time is greater than the slew time, the end of the arc position is used and is calculated as before, but with an angle of (π/2+θ/2).

9.4.3 Sine Wave

The most challenging of the basic trajectories is the sine wave. Figure 9.10 shows the sine reference trajectory for an amplitude of 0.1 m and a length of 0.25 m.

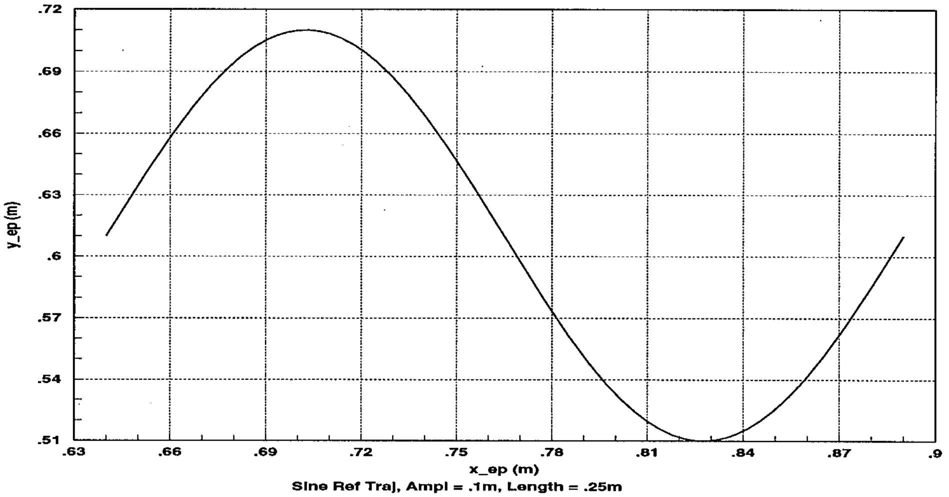


Figure 9.10- Sine Wave Reference Trajectory

By specifying the start position, amplitude, length (period), and slew time, many different sine trajectories are possible. A graphical depiction of the sine trajectory is shown in Figure 9.11. Mathematically, this trajectory can be produced for any desired slew time (except 0). Knowing the starting point (x_1,y_1), the desired amplitude (A), the desired length (B), and the slew time (Tslew), a reference trajectory can be generated from the following equations.

$$\Delta x = B\left(\frac{t}{Tslew}\right) \quad (9.19)$$

$$\Delta y = A\sin\left(\left(\frac{t}{Tslew}\right)2\pi\right) \quad (9.20)$$

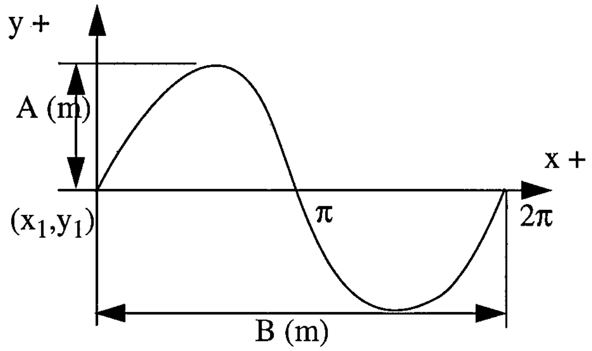


Figure 9.11- Graphical Depiction of the Sine Trajectory

Knowing the change in x and y allows calculation of the desired x and y positions.

$$x = x_1 + \Delta x \tag{9.21}$$

$$y = y_1 + \Delta y \tag{9.22}$$

When t > Tslew, the end of the sine wave is simply $y_f = y_1$, $x_f = x_1 + B$, and this is continuously commanded at the end of the sine wave.

9.5 Robust SANDY Design

Robust controller design allows one controller to perform over a variety of plant conditions. For a flexible robotic manipulator, two areas which are important for robustness are the end-point mass (and therefore, inertia) and the operating point. The end-point mass varies as a payload is picked up or dropped, and the system dynamics change. Also, the position of the links relative to each other affects the dynamics as well. Providing plants with a range of tip masses and operating points allows an optimization that addresses all the given plant conditions simultaneously. Although the SANDY controller is only guaranteed to work for the specific plant cases included in the optimization criterion, experience has shown that all plants in between will also work [46].

For this design, six different plant cases are considered, as shown in Table 9.3. There is a low and high mass plant at each of the three operating points. The plants are linearized about each of the operating points. For each plant in Table 9.3, three variations are required to complete the optimization. These variations are shown in

Table 9.4. For this design problem, one plant variation is developed for a filtered step command input for the first joint angle, and a second plant applies a filtered step command input to the second joint angle. The filters used for the commands are first order lags with a time constant of 1 second to approximate the actual reference input. In addition, white noise inputs are fed directly into the control inputs to excite the high frequency modes for the third plant condition. This third plant is the key to prevent excitation of the flexible modes. Overall, the total number of plants optimized was 18.

Table 9.3 - Robust Plant Conditions

Plant	**Tip Mass** **kg**	**Tip Inertia** **kg-m^2**	**θ_{1nom}** **deg**	**θ_{2nom}** **deg**
1	.1005	.0001	0	90
2	1.768	.0062	0	90
3	.1005	.0001	20	110
4	1.768	.0062	20	110
5	.1005	.0001	-20	70
6	1.768	.0062	-20	70

Table 9.4 - Optimization Plant Variations

Variation	Purpose
1	Optimize for θ_1 error
2	Optimize for θ_2 error
3	Optimize for high frequency (Noise into both control inputs)

The criterion variables are $\dot{\theta}_1$, $\dot{\theta}_2$, error_θ_1, and error_θ_2, respectively. There are two controls (torque_1 and torque_2) and two sensors (θ_1 and θ_2). The criterion variables are used only in the cost function, and can contain variables other than those available from sensors. Thus, the SANDY controller should have two inputs (the sensors) and two outputs (the torques). Values of the nonlinear constraints are a direct constraint on the control covariance of torque 1 for plant 1 of 10 Nm and a constraint on the maximum real part for the closed-loop eigenvalue of -0.2. Also, no weighting is applied to the control penalties R_1 and R_2, and Q_1= diagonal(0,0,1,0) with Q_2 = diagonal(0,0,0,1). The initial guess of the SANDY controller is:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -8 & 0 \\ 0 & 0 & 0 & -9 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} 0.1 & 0 & 8 & 0 \\ 0 & 0.11 & 0 & 9 \end{bmatrix}, \boldsymbol{D} = \begin{bmatrix} 2 & 0 \\ 0 & 2.1 \end{bmatrix} \quad (9.23)$$

This is a fourth-order controller with 2 integral states and a first-order filter for the θ_1 and θ_2 sensor inputs, and is similar to the PID controller described in a previous chapter. The integral states are added to reduce steady-state error in tracking the joint angles. The optimized variables are A(3,3), A(3,4), A(4,3), A(4,4), B(3,1), B(3,2), B(4,1), B(4,2), C(1,3), C(1,4), C(2,3) and C(2,4) for the first-order filters. In addition, C(1,1),C(1,2),C(2,1), and C(2,2) are the design variables for the integral gains. Finally, the entire D matrix is optimized for the proportional gains.

The final design weightings for each plant case are chosen as:

$$Q1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, Q2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix}, Q3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad (9.24)$$

while no weightings were applied to the control penalty matrices. In addition, a constraint on the maximum real part of the closed-loop eigenvalues was applied to each plant. This maximum real value of the roots was -0.2, and forced the integral poles away from the imaginary axis.

After 2454 iterations, the resulting SANDY controller converged to the following solution:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1.56 & -22.23 \\ 0 & 0 & 0.63 & -5.18 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 4.97 & 6.04 \\ 1.75 & -4.85 \end{bmatrix} \quad (9.25)$$

$$C = \begin{bmatrix} 0.68 & 1.71 & 4.64 & 24.40 \\ 0.38 & 1.40 & 0.60 & 9.91 \end{bmatrix}, D = \begin{bmatrix} 0.76 & -8.82 \\ 0.92 & 9.88 \end{bmatrix}$$

The total CPU time used was 3 hours and 31 minutes on a Digital Equipment Corporation alpha machine.

The bandwidth plots (including the 1st order command prefilter at 1 rad/s) are shown in Figure 9.12 and Figure 9.13. Once again, the filtered plots are shown since that is what is actually fed into the arm and since that same filter value was used in the SANDY optimization. The filtered shoulder motor bandwidth is about 1.25 rad/s while the elbow motor bandwidth is about 1.1 rad/s. Compared to the original SANDY design, the shoulder motor bandwidth is about 0.25 rad/s higher with the robust SANDY design.

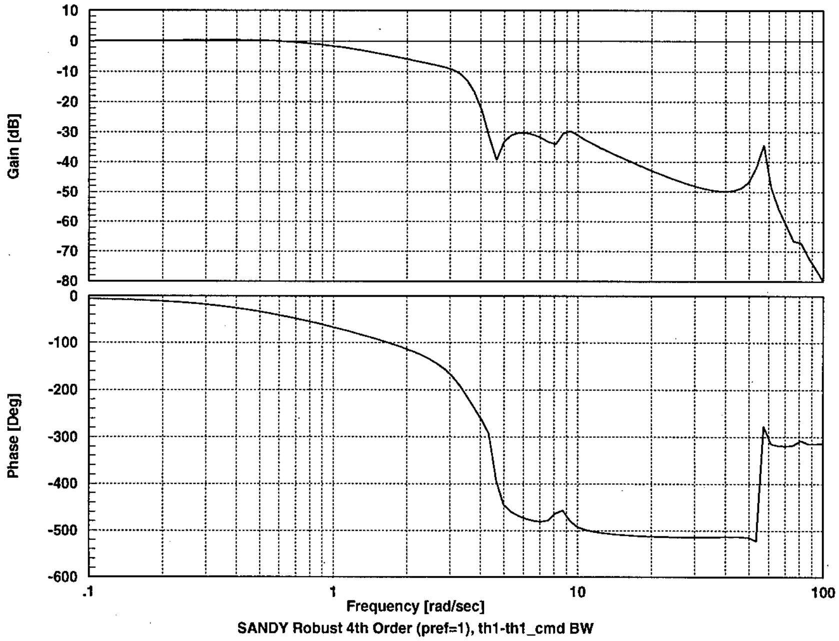


Figure 9.12- Filtered Shoulder Bandwidth, Robust 4th order SANDY

The controller is discretized using a Tustin transformation at a 1 ms sampling time. The difference between a 1 ms simulation and a continuous simulation was negligible for this controller, verifying the initial assumption that a continuous SANDY design should provide a good digital controller at this 1000 Hz sampling rate. If digital considerations are desired, more stringent constraints may be placed on the control covariances to keep the gain on the controller low making it more applicable for higher sampling times (and thus lower sampling rates). Additionally, a direct digital synthesis method using SANDY is being developed.

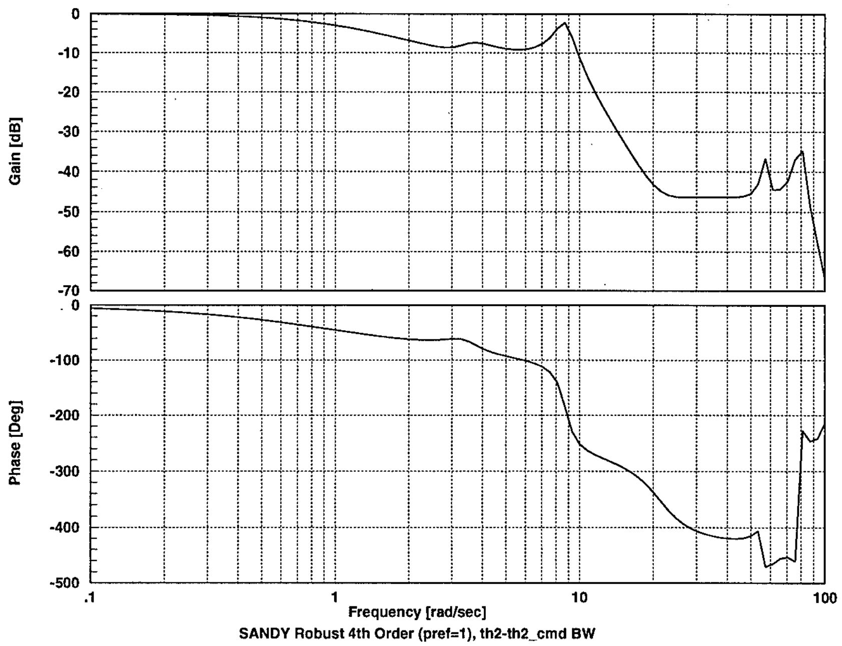


Figure 9.13- Filtered Elbow Bandwidth, Robust 4th order SANDY

9.6 Experimental Comparison

On all the experimental plots showing volts, the conversion factor to Nm is 1/1.9208 for τ_1 and 1/8.045 for τ_2. Please note that all plots in this paper have time units of seconds and have angle outputs measured relative to the starting position.

The medium tip mass nominal configuration was tested first for all three endpoint tasks. All plots compare the 4th-order Robust SANDY and 20th-order LQQ controllers to the reference command. Figure 9.14 shows a 3 sec line slew corresponding to about -20 degree changes in both angles simultaneously. In this plot, the 4th-order SANDY controller matches the reference line slightly worse than the 20th-order LQG controller.

Figure 9.15 shows the same reference line, but allows 10 seconds to perform the maneuver. In this case, both controllers track the reference much better and with comparable performance.

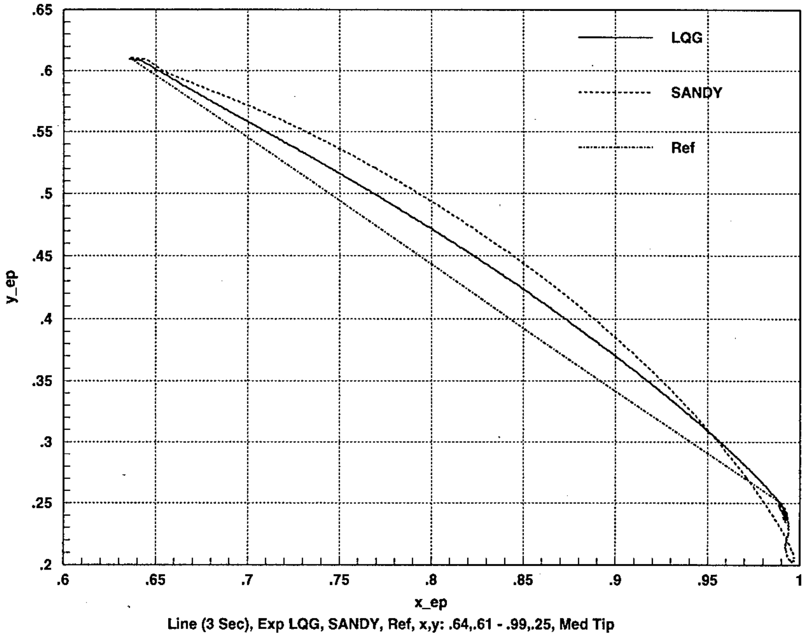


Figure 9.14- Exp 3 Sec Line: LQG, SANDY, Ref from x,y=.64,.61 - .99,.25 m

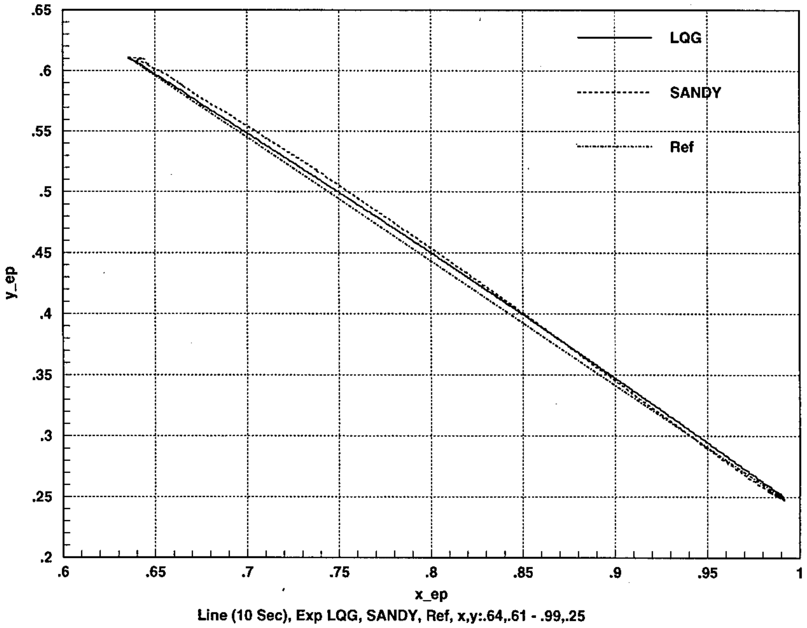


Figure 9.15- Exp 10 Sec Line: LQG, SANDY, Ref from x,y=.64,.61 - .99.25 m

Another end-point task is tracking an arc. In this case, a 40 degree arc with a radius of curvature of 0.51 inches is tracked in 3 seconds. Notice that the LQG controller tracks much more accurately for this case, as shown in Figure 9.16.

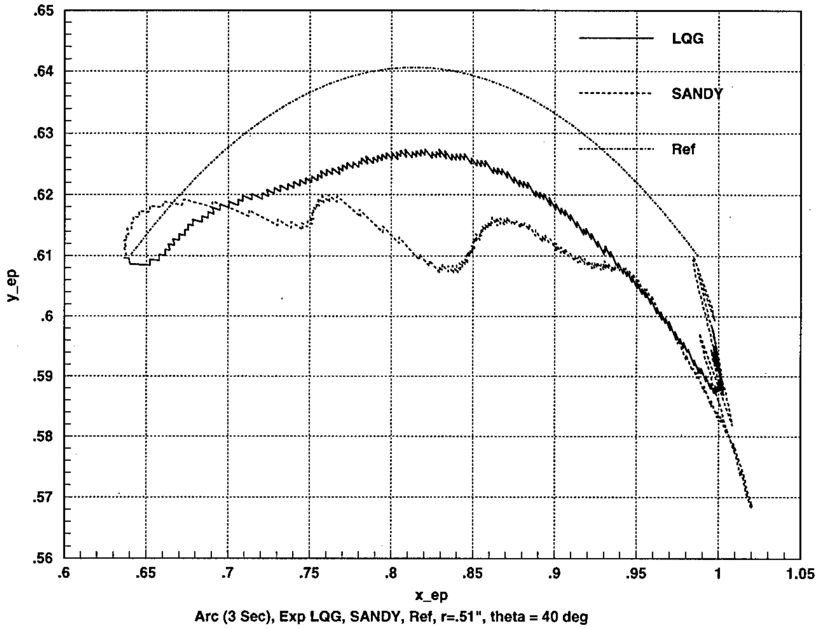


Figure 9.16- Exp 3 Sec Arc: LQG, SANDY, Ref for R=.51", θ=40 deg

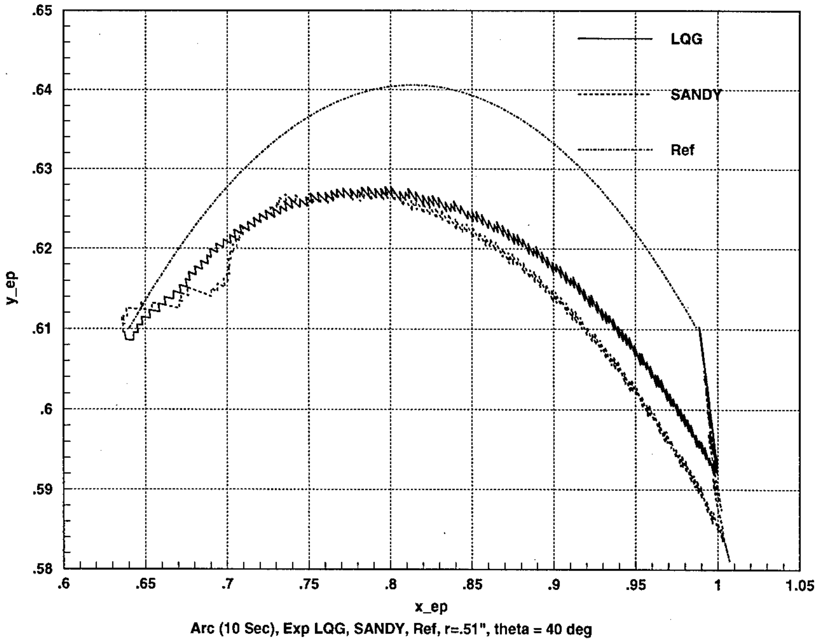


Figure 9.17- Exp 10 Sec Arc: LQG, SANDY, Ref for R=.51", θ=40 deg

The reason for the poor SANDY performance in this case is the low gains for the first motor and the relatively little amount of traverse required for the first motor by this trajectory. Thus, only small torques are required and they don't overcome the residual friction in the motor shaft initially. Note that the steady-state error is lower for the SANDY controller, though.

In Figure 9.17, notice that the performance between the controllers is much closer, even though the LQG controller still does a slightly better job. Thus, the tracking performance for both controllers increases as the task is done more slowly.

Figure 9.18 shows that both controllers have trouble tracking a sine wave with an amplitude of 0.1 m and a length of 0.25 m in 3 seconds. The SANDY controller produces more overshoot, but the tracking trends are close for both.

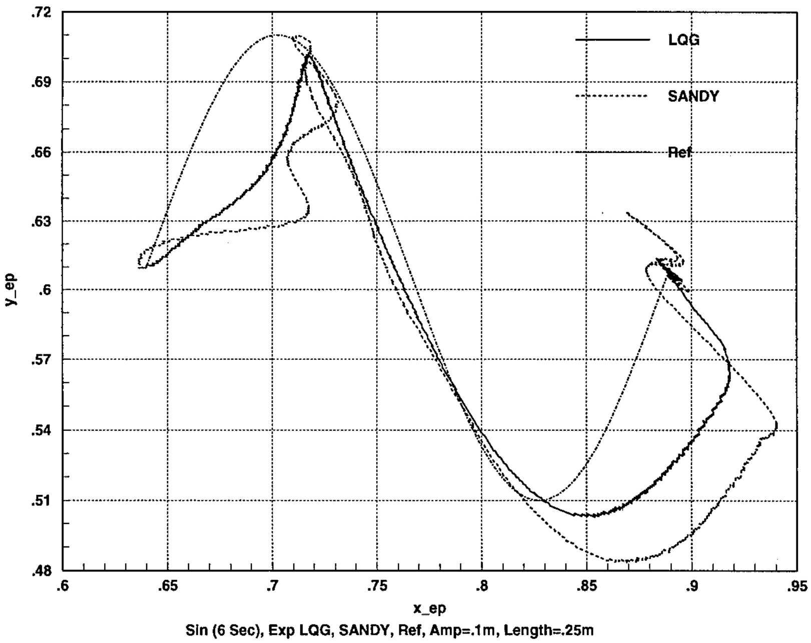


Figure 9.18- Exp 6 Sec Sine: LQG, SANDY, Ref for Amp=.1m, Length=.25m

By increasing the tracking time to 20 seconds, significant improvement for both controllers occurs. The SANDY controller initially has several small oscillations, but has lower steady-state error. Figure 9.19 shows the results of this trajectory.

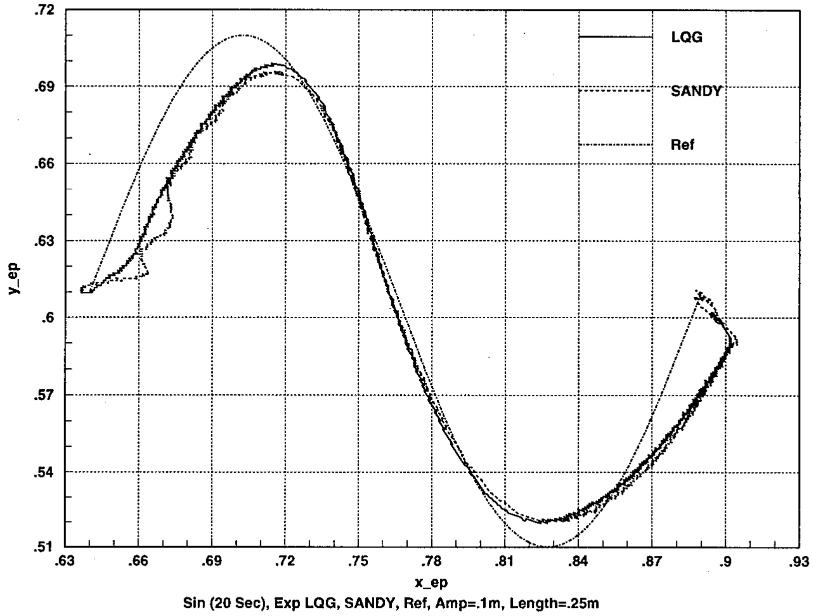


Figure 9.19- Exp 20 Sec Sine: LQG, SANDY, Ref for Amp=.1m, Length=.25m

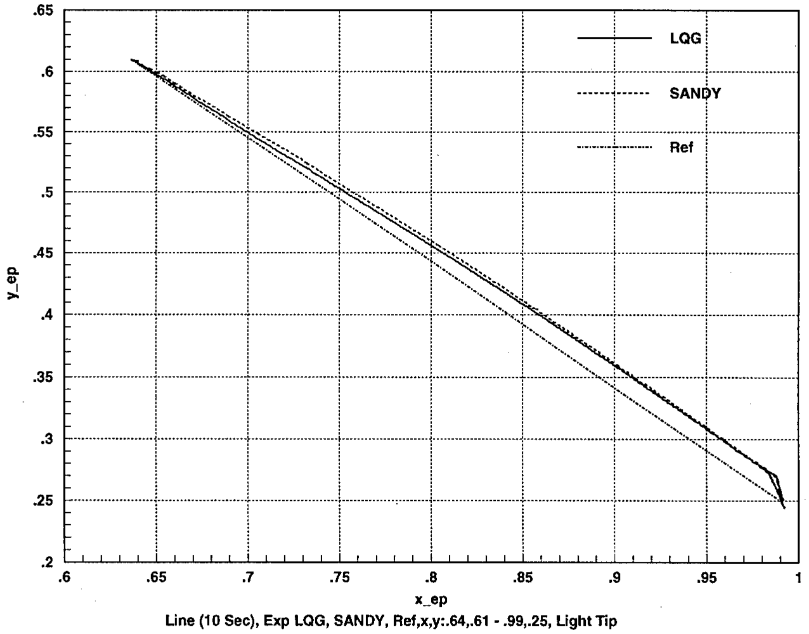


Figure 9.20- Exp 10 Sec Line: LQG, SANDY, Ref from x,y=.64,.61 - .99.25 m, Light Tip

The different trajectories test the robustness to arm position, but not to payload uncertainty. Therefore, tests are run with the light tip and a heavy tip used in the robust SANDY design. The same line trajectory is run in 10 seconds with the light tip and the results are shown in Figure 9.20. In this case, both controllers are comparable.

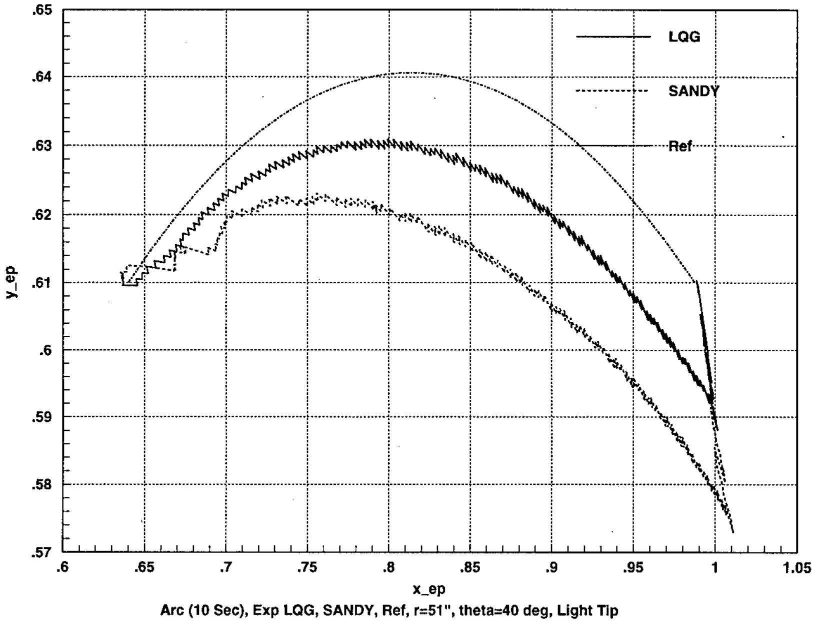


Figure 9.21- Exp 10 Sec Arc: LQG, SANDY, Ref for R=.51", θ=40 deg, Light Tip

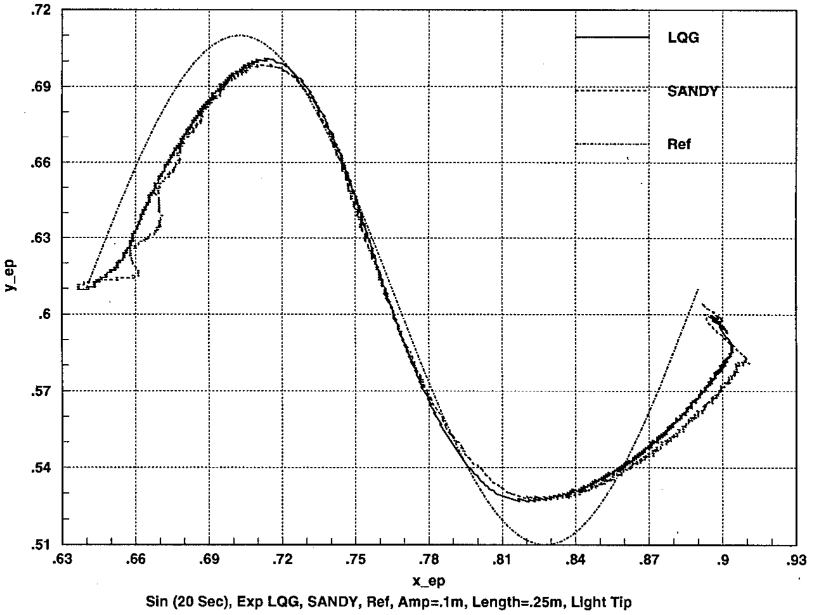


Figure 9.22- Exp 20 Sec Sine: LQG, SANDY, Ref for Amp=.1m, Length=.25m, Light Tip

When the arc trajectory is run in 10 seconds with the light tip, the LQG controller still provides better performance than the SANDY controller, as shown in Figure 9.21.

The same trends as with the medium tip occur for the 20 second sine trajectory. Figure 9.22 shows that both controllers provide comparable performance

Switching to the heavier tip with a ball and cup provides a payload test in the opposite direction to the light-weight tip. Figure 9.23 shows that for the 10 sec line trajectory, the SANDY controller provides better performance than the LQG controller. In fact, the SANDY controller shows little degradation for this trajectory despite speed and payload variations.

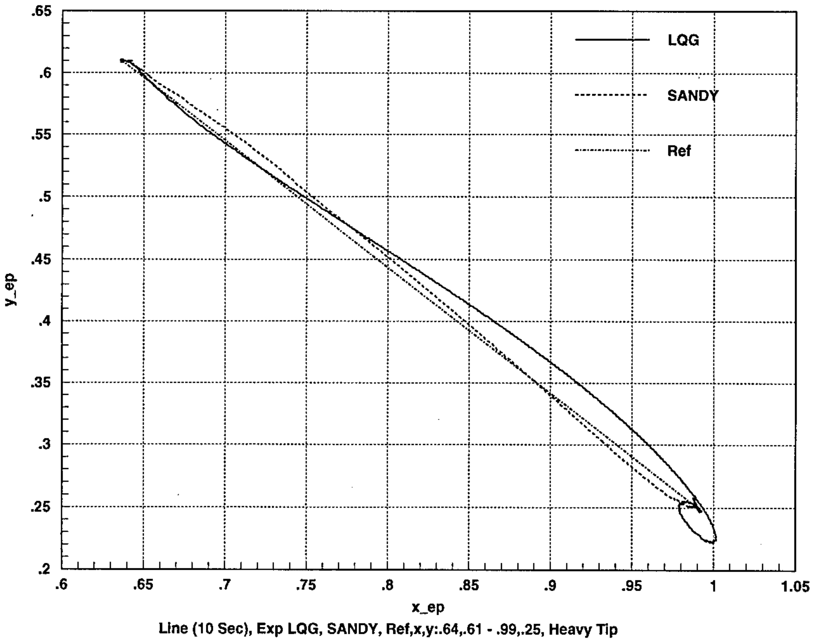


Figure 9.23- Exp 10 Sec Line: LQG, SANDY, Ref from x,y=.64,.61 - .99.25 m, Heavy Tip

For the 10 second arc trajectory with the heavy tip, the LQG performance actually improves, while the SANDY controller degrades, as shown in Figure 9.24.

Finally, the performance for both controllers is fair for the 20 second sine trajectory, which Figure 9.25 shows. The LQG controller has a slight edge here, as the SANDY controller initially has a lag in following the sine wave with a heavier mass, probably due to the low first motor control gains again.

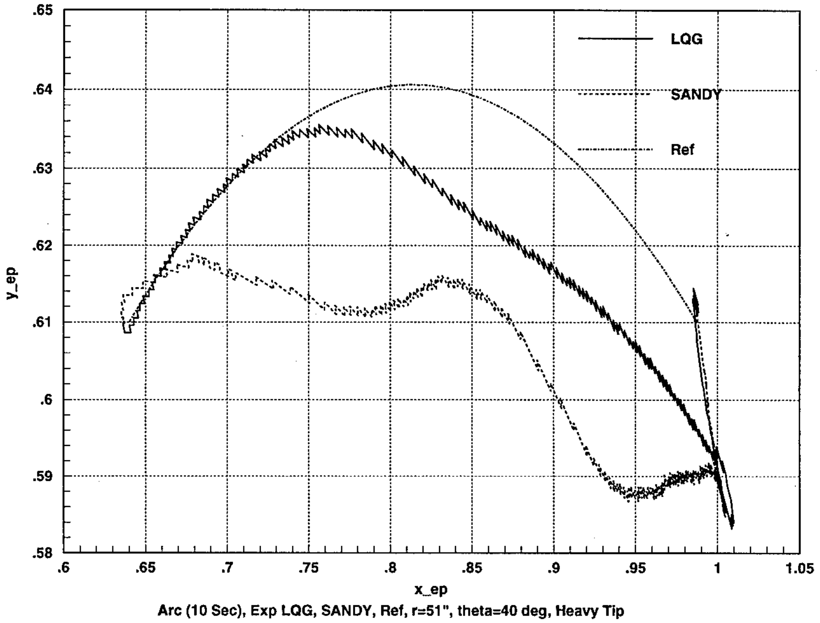


Figure 9.24- Exp 10 Sec Arc: LQG, SANDY, Ref for R=.51", θ=40 deg, Heavy Tip

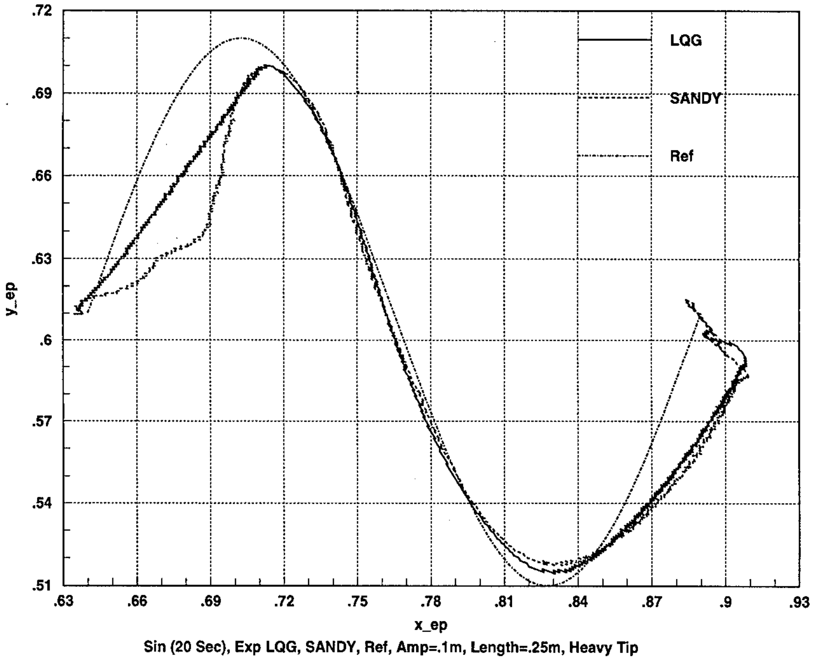


Figure 9.25- Exp 20 Sec Sine: LQG, SANDY, Ref for Amp=.1m, Length=.25m, Heavy Tip

Overall, the SANDY controller provides better performance for the line trajectory, the LQG controller performs better for the arc trajectory, and both are about the same for the sine trajectory. Both controllers proved robust to payload and arm position, but both exhibited degraded performance when the tracking speed increased.

Appendix D contains additional plots for varying speeds along the line and arc trajectories. The performance improved as the speed decreased, and degraded as the speed improved.

10 Hybrid Position and Force Control Overview

A challenging real-world application for a flexible manipulator is applying a constant force to a surface. Examples of this application include buffing, grinding, polishing, or even painting surfaces. Accomplishing such tasks requires an algorithm which controls both position and force simultaneously.

Humans apply position control with force feedback when trying to accomplish such tasks as placing a peg in a hole. The eye provides an estimate of the position, and the nerves in the hand and arm act as force sensors. For a robotic manipulator, the "eyes" of the system are the position sensors and the "nerves" of the system are the components of force sensed from a 6-component force/torque sensor. When humans follow a surface, the eyes allow them to place a hand onto a surface, and the nerves provide feedback as to how much force is applied to the surface. In the same manner, a robot can use both force and position information to follow a surface.

For a rigid surface, like metal or wood, the force and position controllers can be decoupled [34]. The force control is used to apply torques in the normal direction, which is also known as axial or longitudinal control. The position control operates on the tangential or lateral direction. Thus, the two controllers can be decoupled and designed separately. The overall hybrid control is thus position control along the surface tangent and force control along the surface normal.

10.1 Force Control

One of the two required tasks for surface following is force control. The lumped-spring-mass-damper model used for position control does not address any of the force characteristics of the TLA. Therefore, a force model is required. Additionally, a way to generate the desired force reference commands is required. Before the force model is developed and the algorithm explained, some basic concepts behind force control are needed.

10.1.1 Jacobian

It is appropriate to describe the Jacobian as it relates to robotic applications [15].

Given a set of equations

$$Y = F(X) \tag{10.1}$$

where

$$y_i = f_i(x_1, x_2, \ldots, x_n) \tag{10.2}$$

and

$$Y = [y_1, y_2, \ldots, y_n]^T \tag{10.3}$$

To calculate the differentials of y_i as a function of the differentials of x_j, use the chain rule to obtain

$$\begin{aligned} \delta y_1 &= \frac{\partial f_1}{\partial x_1}\delta x_1 + \frac{\partial f_1}{\partial x_2}\delta x_2 + \ldots + \frac{\partial f_1}{\partial x_n}\delta x_n \\ \delta y_2 &= \frac{\partial f_2}{\partial x_1}\delta x_1 + \frac{\partial f_2}{\partial x_2}\delta x_2 + \ldots + \frac{\partial f_2}{\partial x_n}\delta x_n \\ \delta y_n &= \frac{\partial f_n}{\partial x_1}\delta x_1 + \frac{\partial f_n}{\partial x_2}\delta x_2 + \ldots + \frac{\partial f_n}{\partial x_n}\delta x_n \end{aligned} \tag{10.4}$$

which simplifies to

$$\delta Y = \frac{\partial F}{\partial X}\delta X \tag{10.5}$$

The matrix of partial derivatives is known as the Jacobian, denoted by **J**. Thus, another expression which is equivalent to Equation (10.5) is

$$\delta Y = J(X)\,\delta X \tag{10.6}$$

For the TLA shown in Figure 10.1, the Jacobian becomes

$$\begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \begin{bmatrix} (-d_1 \sin\theta_1 - d_2 \sin(\theta_1 + \theta_2)) & (-d_2 \sin(\theta_1 + \theta_2)) \\ (d_1 \cos\theta_1 + d_2 \cos(\theta_1 + \theta_2)) & d_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \end{bmatrix} \tag{10.7}$$

Notice that the Jacobian varies with the manipulator position. For example, the mapping between angles and end-point is quite different between the (0°,90°) configuration and the (0°,0°) configuration, where the numbers in parenthesis are (θ_1,θ_2). At the first condition, a small change in θ_2 is almost exclusively a change in x, while at the second condition a small change in θ_2 is almost exclusively a change in y.

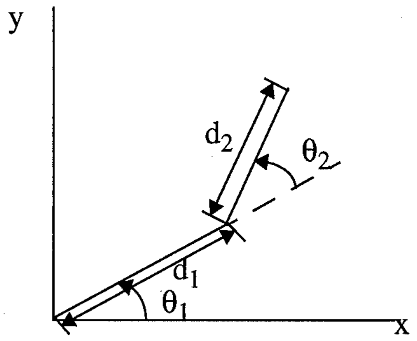


Figure 10.1- Jacobian Parameters for the TLA

10.1.2 Principle of Virtual Work

The relationship between torque and force is critical for force control since the force is controlled through the joint torques. When forces act on a device, work is done if the device goes through a displacement [15]. By definition, work is force acting though a distance. The principle of virtual work allows relationships for the static case as the displacement goes to an infinitesimally small amount. For a multi-link manipulator, this principle may be written in equation form as

$$\boldsymbol{T}^T\delta\theta - \boldsymbol{F}^T\delta x = 0 \tag{10.8}$$

where for the TLA

$$\boldsymbol{T} = \begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} \tag{10.9}$$

$$\delta\theta = \begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} \tag{10.10}$$

$$\delta x = \begin{bmatrix}x\\ y\end{bmatrix} \tag{10.11}$$

and **T** is a vector of joint torques, δθ is a vector of joint angles, and **δx** is a vector of end-point position coordinates.

For the TLA, we can write the relationship between the end-point and the angles as

$$\delta x = J\delta\theta \tag{10.12}$$

10.1.3 Identification of the Force Transfer Functions

Before design and simulation of a force controller may begin, a model is required. One experimental method which may identify the force transfer functions is Approximate Linearization [34]. Using the Jacobian, one can calculate the joint torques necessary to provide a desired end-point force for the case of static equilibrium. Thus, feeding a sinusoid into a joint while positioning the end-point at a solid object allows for a sinusoidal application of force at the end-point. Thus, a frequency response curve for torque at joint 1 to the axial force as well as torque at joint 2 to the tangential force can be obtained. From the frequency response, a modal representation of the system may be obtained in the form of a transfer function which matches the experimental frequency response. Figure 10.2 shows a graphical depiction of the method.

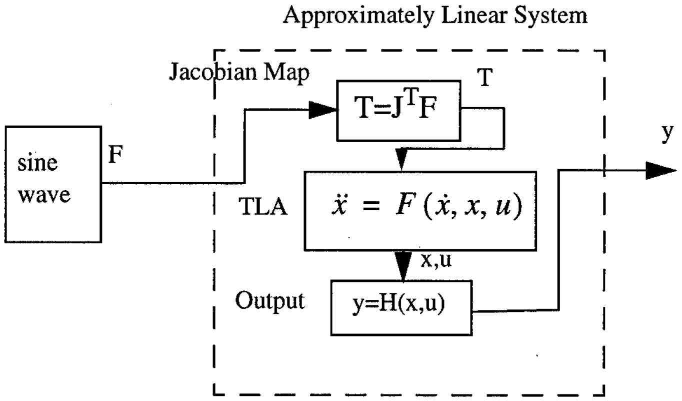


Figure 10.2- Identification by Approximate Linearization

With an understanding of the Jacobian and the Principle of Virtual Work, the Approximate Linearization makes sense. To verify the position model, a torque is applied as a sine wave to the motor at varying frequencies. A transfer function between the commanded angle and measured angle is obtained at different frequencies to determine the frequency response. Since torque is related to force through the Jacobian [34], it makes sense that a frequency response for force should be obtained in a similar manner to the position transfer function, which depends on the torque directly. Another way to think of the relationship between the position and force transfer functions is that the position transfer function corresponds to an end-point which is unpinned, while the force transfer function corresponds to an end-point which is pinned.

10.1.4 Force Components and Coordinates

Since the force controller is outfitted with a low-friction roller, essentially the only force sensed at the surface is the normal component of force [34]. There is a relationship between the surface normal and the surface tangent force, which is shown in Figure 10.3.

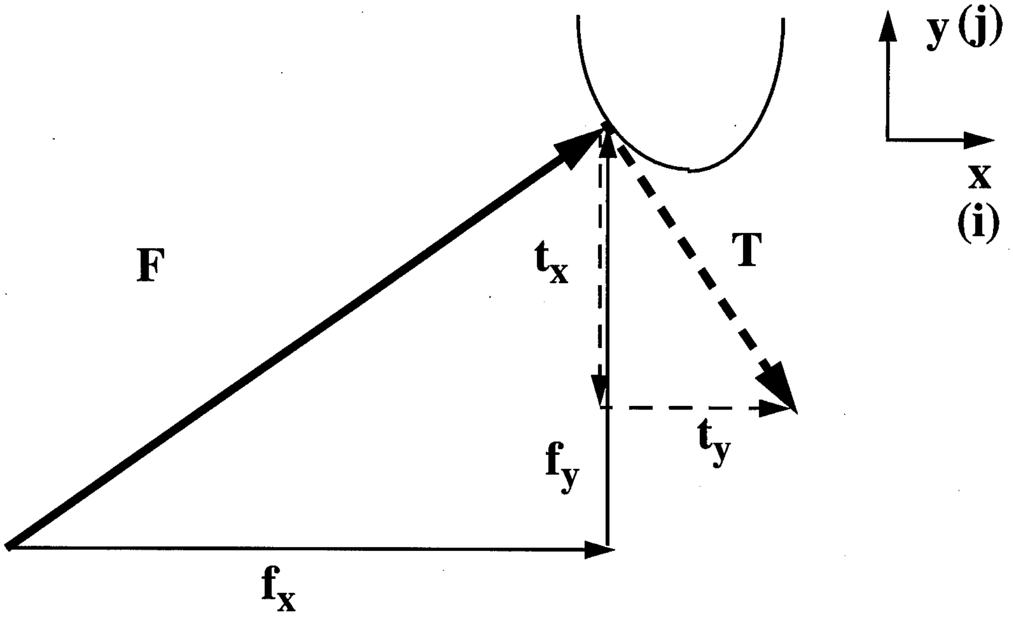


Figure 10.3- Surface Force Diagram

The surface normal, which is the sensed force, has both x and y components. These are related to the tangential direction components through similar triangles. Both can be represented in equation form [34].

$$F = f_x\hat{i} + f_y\hat{j} \tag{10.13}$$

where **F** is the surface normal force. Additionally, the tangential direction vector **T** can be represented as

$$T = t_x\hat{i} + t_y\hat{j} \tag{10.14}$$

Finally, the tangential components are related to the normal components by the following equation

$$T = \left(\frac{f_y}{|F|}\right)\hat{i} - \left(\frac{f_x}{|F|}\right)\hat{j} \tag{10.15}$$

Another consideration when examining force components is the reference frame. The desired forces are normally considered in the inertial, or fixed reference frame. The frame which is associated with the force transducer is defined along the second link.

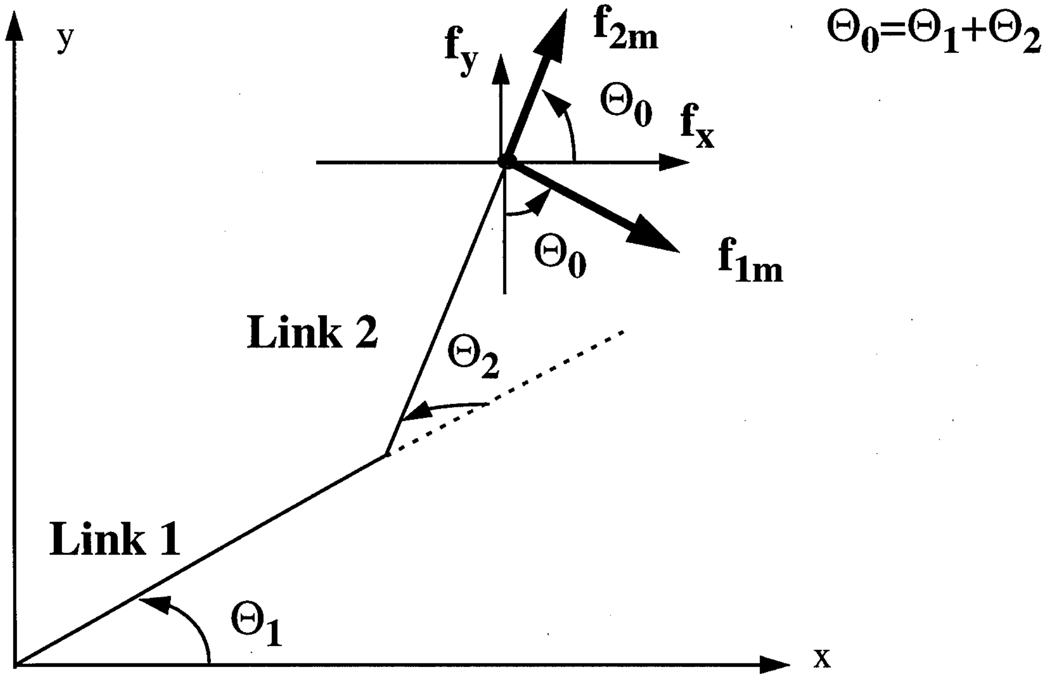


Figure 10.4- Relationship Between Inertial and Measured Frames

This frame is called the measured frame, since the force measurements take place here. Figure 10.4 shows the relationship between the two frames. Notice the relationship between the two frames is simply a function of the combination of the first and second link angles. Mathematically, the relationship is expressed as:

$$f_x = f_{1m}\sin(\theta_0) + f_{2m}\cos(\theta_0) \tag{10.16}$$

and

$$f_y = -f_{1m}\cos(\theta_0) + f_{2m}\sin(\theta_0) \tag{10.17}$$

where

$$\theta_0 = \theta_1 + \theta_2 \tag{10.18}$$

The conversion from the inertial frame to the measured frame is also needed. The equations are:

$$f_{xm} = f_x \sin\theta_0 - f_y \cos\theta_0 \tag{10.19}$$

and

$$f_{ym} = f_x \cos\theta_0 + f_y \sin\theta_0 \tag{10.20}$$

10.1.5 Force Control Structure

With knowledge of the basic concepts covered in the previous force sections, a force control scheme becomes plausible. Figure 10.5 shows a block diagram of the scheme. In words, the x and y components of force arm are obtained in the measured frame, and a magnitude is calculated. Using the angle positions for the links, a conversion to the inertial frame is accomplished. Once in the inertial frame, the normal components for force are calculated. These components can be thought of as directions of the force, and are combined with the desired magnitude to produce reference commands for both the x and y components of force in the inertial frame. Since the actual forces are obtained in the measured frame, the reference commands are converted to the measured frame. An error signal in both components of force is produced, and fed into the force controller. The resulting force commands are then converted to joint torque commands through the Jacobian. It is these joint torque commands that are fed to motors to move the arm.

The objective of this portion of the design is determine a controller which provides high bandwidth force control without exciting any of the arm flexible modes. The same general scheme is used regardless of the specific controller.

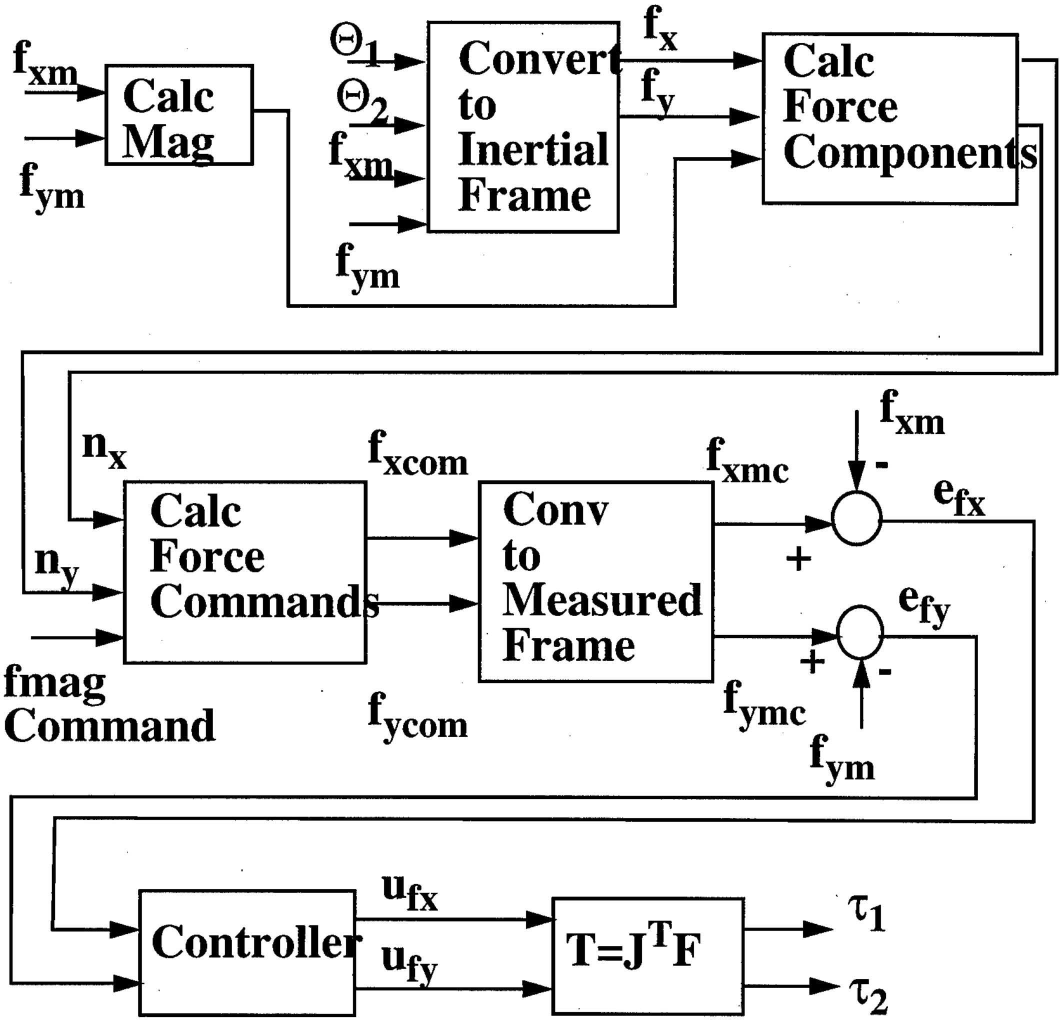


Figure 10.5- Force Control Scheme

10.2 Position Control

For a rigid surface, the lateral or tangential direction is primarily controlled by the position controller. In addition, the position controller is the only controller used to move the arm until contact is made with a surface.

10.2.1 Position Control With Surface Knowledge

In many industrial applications, the surface shape is known. In this case, the position control scheme is almost exactly like that described in Chapter 9. The reference trajectory in terms of end-point is generated to match the trajectory from the starting position of the arm to the surface. Next, the surface contour coordinates are used as the position reference for the remainder of the trajectory. Essentially, the x and y splines are replaced by a more complex trajectory generator, but everything else remains the same for position control.

10.2.2 Position Control Without Surface Knowledge

The most challenging control application for surface following with force control is to follow a surface without knowledge of the surface. One surface following algorithm which tackles this problem was used by Kraft [34]. Figure 10.6 shows a picture of this algorithm.

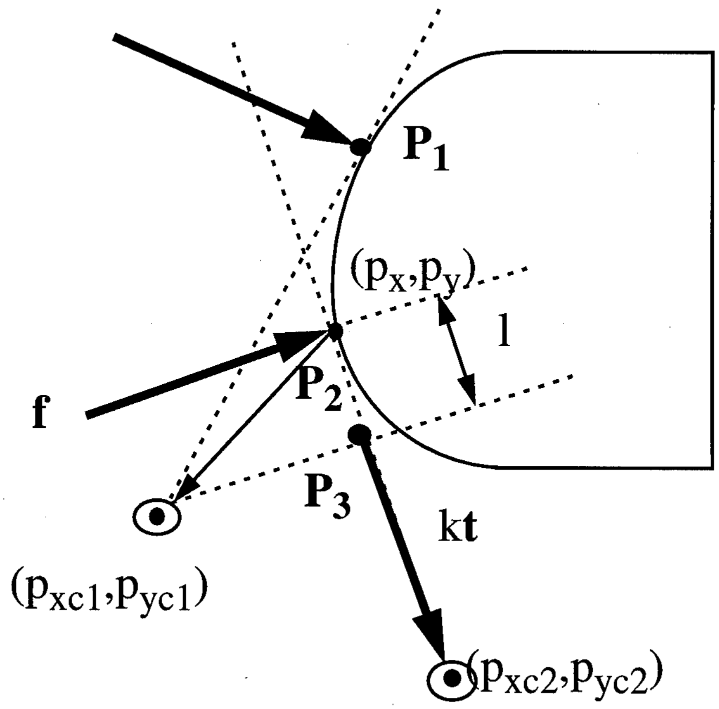


Figure 10.6- Surface Following Algorithm

The surface following algorithm can be described both in words and by equation. In words:

1. Define the end-point to be at point P_1 at time t_0.

2.a. Define a position in Cartesian coordinates (p_{xc1},p_{yc1}). These coordinates are located a distance k along the tangent direction.

2.b. At the same time as 2.a., a force in the direction opposite the surface normal is commanded.

3. At time t_1, one sample period after t_0, the end-point is at P_2 having the Cartesian coordinates (p_x,p_y). The new surface normal and tangent directions are calculated at this new position.

4. The previous commanded position, (p_{xc1},p_{yc1}), is then projected onto a line that is parallel to the new surface tangent and passes through P_2. This defines the point P_3.

5.a. The next commanded position, (p_{xc2}, p_{yc2}), is then located a distance k from P_3 along the new tangent direction.

5.b. Again, a force in the direction opposite the current surface normal is commanded at the same time as 5.a..

This process is repeated every sample period. The resulting surface tracking velocity can then be expressed as

$$v_{tracking} = \frac{k}{T} \tag{10.21}$$

where T is the sample period.

The algorithm may also be stated in a more mathematical form. Defining the normal and tangential forces in Equations (10.13)- (10.15) is the starting point for the mathematical description. Next, let (p_x,p_y) be the Cartesian coordinates of the current end-point position, and let (p_{xc1},p_{yc1}) be the Cartesian coordinates of the last commanded position. Then the projection of the last commanded position on a line passing through (p_x,p_y) and parallel to **t** is

$$l = (p_{xc1} - p_x)\, t_x + (p_{yc1} - p_y)\, t_y \tag{10.22}$$

Also, the Cartesian coordinates of the next commanded position, (pxc2,pyc2), are given by

$$p_{xc2} = p_x + (l + k)\,t_x \tag{10.23}$$

and

$$p_{yc2} = p_y + (l + k)\,t_y \tag{10.24}$$

where k is a constant.

10.3 Hybrid Control Structure

The hybrid control structure involves torques from the position controller and the force controller. Figure 10.7 shows a block diagram of the hybrid scheme.

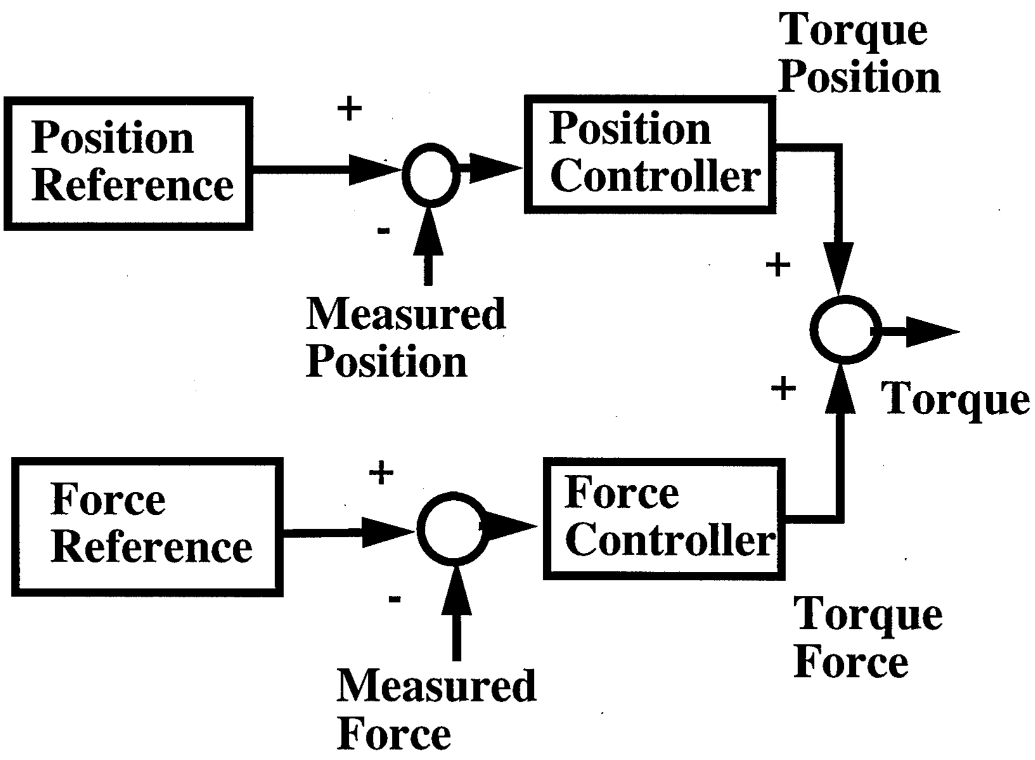


Figure 10.7- Hybrid Force and Position Control Scheme

It is important to realize that in this scheme, the two controllers may oppose each other in some circumstances. For example, if the commanded position is away from the surface, then the position controller will try to move the end-point away from the surface. At the same time, the force controller will try to maintain a constant force by moving the end-point into the surface. Thus, the two controllers are both trying to accomplish the opposite effect! One way to reduce this effect is to bandwidth-separate the controllers so that one controller tends to dominate the response [34]. In this case, force control should have a higher bandwidth than the position controller.

One other insight into this particular application is that generally, the position controller will control the lateral direction and the force controller will control the longitudinal direction, as stated earlier. Thus, if the surface is known, the position command will generally not command the end-point to leave the surface. For the unknown surface case, the algorithm should command a direction along the tangent and not into surface. Therefore, conditions where the two controllers are fighting each other should be minimized.

10.4 Task Overview

The task overview can be stated as:

> To develop robust reduced-order position and force controllers that allow surface following of an unknown surface while maintaining a constant contact force. The control task starts from any arbitrary initial location of the flexible link manipulator and uses position control until touchdown on the object, then uses hybrid position and force control to perform the surface following. Finally, the task ends by returning to the initial arbitrary starting location.

This represents a typical industrial application such as polishing or grinding. Once the framework is designed, the task is to design the best controllers. Figure 10.8 shows a diagram of the surface following task. Notice that task requires movement of the end-point in two distinct phases - pure position movement from 1-2, 2-3 and 4-1, and then hybrid force and position control from 3-4 while following the surface. The most critical part of the task is the touchdown on the surface, which occurs at point 3. The control of the end-point is achieved through movement of the actuators for links 1 and 2.

Since this is a flexible manipulator, the end-point position can only be approximated through forward kinematics. However, the force control uses a non-collocated sensor, namely the force sensor. Thus, the end-point position need not be known exactly since the force sensor compensates for any uncertainty.

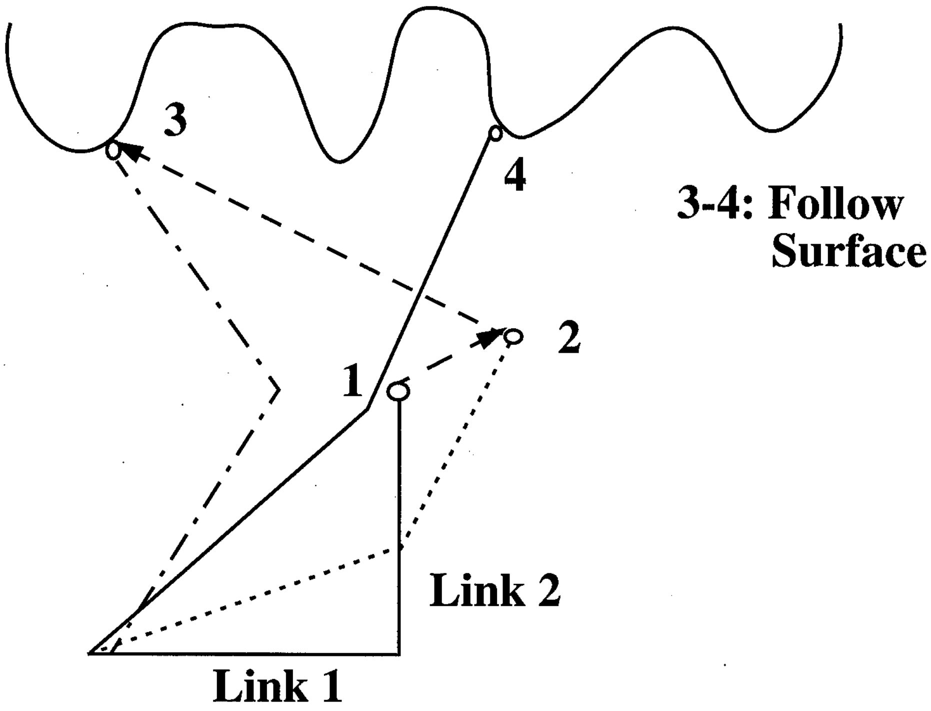


Figure 10.8- Surface Following Task

Currently, two surfaces are available as test cases for the surface following tasks. These surfaces are shown in Figure 10.9. The simplest surface to follow is a straight line, hence this target provides a baseline to test the controllers initially. The arc is more challenging since it requires some interaction between the two links, and can be placed in either a concave or a convex position. It most closely matches a trajectory which might be used in industry. In particular, at the top of the semicircles, the controllers must provides a means to maintain contact even after previous movement of the end-point in a trajectory which might lead off the surface.

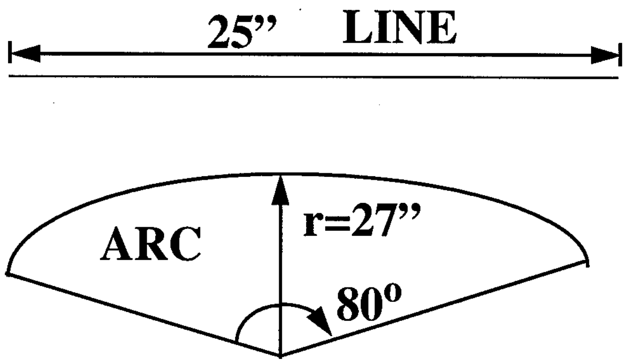


Figure 10.9- Available Surfaces

11 Force Control Model and Design

The ultimate goal of this research is to provide path following for a two-link flexible robotic manipulator through the use of hybrid position and force control. Since these controllers act in perpendicular directions for a rigid surface [34], it is logical to design them separately. The position control designs are discussed in previous chapters, so this chapter will focus on the design of force controllers. Before the force controllers can be designed, however, force models are needed for both the x and y directions.

11.1 Force Models

The force models are based on the approximate linearization technique used by Kraft [34]. This is an empirically-based technique which maps the motor torques to forces through the Jacobian, as shown in Chapter 10. Sine sweeps are done separately for the x-force and y-force. When conducting the y-force (normal force) test, the sensor is brought in contact with surface perpendicular to the y direction. In a like manner, the x-force (side or tangential force) test is conducted against a surface perpendicular to the x direction. Then, the frequency responses of the tests are matched to mathematical models for both axes, and thus two force models are obtained. For this work, both the x and y models are 11th order single-input single-output transfer functions. The model consists of a filter combined with different modes to best match the frequency response. In addition, there is a noise filter at 10 rad/s on the force transducer which is implemented in the data acquisition software.

The transfer function for the Fy model is:

$$\frac{fy}{fycmd} = 0.525\left(\frac{10}{s+10}\right)\left(\frac{60}{s+60}\right)\left(\frac{80}{s+80}\right)\left(\frac{3364}{s^2+0.1s+3364}\right)\left(\frac{22500}{s^2+3s+22500}\right)\left(\frac{78400}{s^2+16.8s+78400}\right)\ldots$$

$$\left(\frac{1.2x10^5}{s^2+3.5s+1.2x10^5}\right)\left(\frac{s^2+80s+6400}{6400}\right)\left(\frac{s^2+420s+9x10^5}{9x10^5}\right)\left(\frac{s^2+525s+1.4x10^5}{1.4x10^5}\right) \quad (11.1)$$

The transfer function for the Fx model is:

$$\frac{fx}{fxcmd} = 0.58\left(\frac{30}{s+30}\right)\left(\frac{16.8}{s^2+5.7s+16.8}\right)\left(\frac{5.0x10^3}{s^2+0.7s+5.0x10^3}\right)\left(\frac{s^2+11s+30}{30}\right)\left(\frac{5.1x10^4}{s^2+5.7s+5.1x10^4}\right)\ldots$$

$$\left(\frac{1.1x10^{5}}{s^{2}+22.5s+1.1x10^{5}}\right)\left(\frac{s^{2}+4.2s+44000}{44000}\right)\left(\frac{s^{2}+300s+63000}{63000}\right)\left(\frac{25600}{s^{2}+50s+25600}\right) \quad (11.2)$$

11.2 Force Control Designs

As a basis of comparison, standard linear-quadratic-gaussian (LQG) and proportional-integral (PI) controllers are used. Next, the LQG controllers are reduced through both balanced truncation and modal residualization. The key force control design method is the SANDY direct optimization program. Using SANDY allows the design of robust reduced-order force controllers, which is another key thrust of this dissertation. Since there is an x and y force model, separate force controllers must be designed for each direction.

11.2.1 PI Force Control

One classical control approach which is used due to its simplicity is PI control. For this application, the PI control gains were iteratively tuned in simulation and then experimentally. The Fx controller uses gains of PI=.3,5 and the Fy controller uses gains of PI=.7,6. Figure 11.1 shows the experimental and simulation results for the Fx controller match fairly well. Similarly, Figure 11.2 shows the same results for the Fy controller which also displays a close match between experimental and simulation.

11.2.2 LQG Force Control

For this design, the LQG controller uses the same general procedure as described for LQG position control. In this case, however, there are two distinct controllers - one for Fx and one for Fy. Also, there is 1 reference input, 1 sensor input, and one control output. Both Fx and Fy models are 11th order, with a noise filter in each sensor loop located at 10/(s+10). Thus, the plant is effectively 12th order. Additionally, an integral control state is added to both controllers to reduce steady state error. Therefore, both the LQG Fx and Fy controllers are 13th order.

First, the Fx LQR criterion is simply the error in Fx and the control is simply ufx. So, the LQR weightings for the Fx controllers are Q=40 and R=.075. The LQE estimator assumes no process noise, except for a small amount at the control input. Also, the sensor is the Fx measurement.

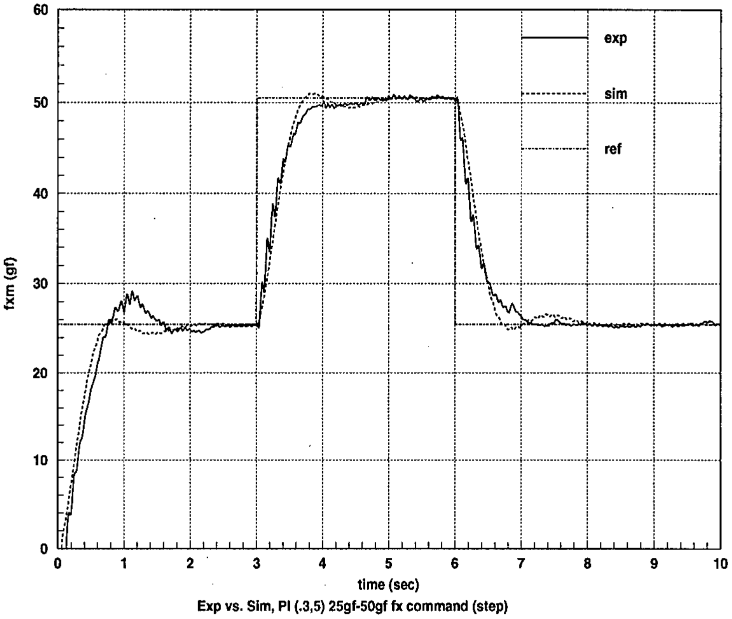


Figure 11.1- Experimental vs. Sim, PI=.3,5 , Fx Controller

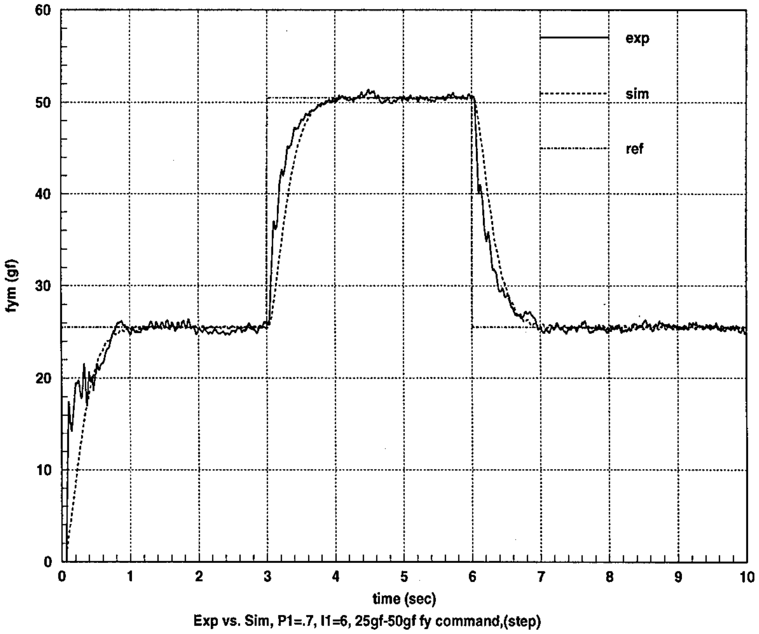


Figure 11.2- Experimental vs. Sim, PI=.7,6 , Fy Controller

The weightings are W=[.01] and V=[1] for this design. The resulting LQG controller is:

$$A = \begin{bmatrix} -18.7 & -12.4 & -1.8 & -9.2 & -3.9 & -7.9 & -4.5 & -8.7 & -5.2 & -14.7 & -16.6 & -8 & 92.4 \\ -10 & -154.4 & -192 & -83.4 & -165.6 & -49.2 & -92.5 & -30.1 & -43.7 & -43 & -14 & -9.3 & 0 \\ 0 & 1024 & 0 & 0 & 0 & 0 & 2e-6 & 5.4e-6 & 1.1e-5 & 6.9e-5 & 4.6e-5 & 3.1e-5 & 0 \\ 0 & 0 & 256 & 0 & 0 & 1.3e-5 & 3.3e-5 & 9.1e-5 & 1.9e-4 & 1.2e-3 & 7.6e-4 & 5.1e-4 & 0 \\ 0 & 0 & 0 & 256 & 0 & -1e-5 & -2.5e-5 & -7.0e-5 & -1.4e-4 & -8.9e-4 & -5.9e-4 & -4.0e-4 & 0 \\ 0 & 0 & 0 & 0 & 256 & -1.7e-4 & -4.3e-4 & -1.2e-3 & 2.4e-3 & -1.5e-2 & -9.9e-3 & -6.6e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 128 & 1.6e-4 & 4.4e-4 & 9.1e-4 & 5.6e-3 & 3.7e-3 & 2.5e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.7e-4 & 128 & 3.9e-3 & 8e-3 & 5e-2 & 3.3e-2 & 2.2e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -9.6e-5 & -2.4e-4 & 64 & -1.3e-3 & 8.3e-3 & -5.5e-3 & -3.7e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2.5e-4 & -6.3e-4 & -1.7e-3 & 32 & -2.2e-2 & -1.5e-2 & 9.8e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -9.7e-5 & -2.4e-4 & -6.6e-4 & -1.4e-3 & 16 & -5.6e-3 & -3.8e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -7e-5 & -1.7e-4 & -4.7e-4 & -9.7e-4 & -6e-3 & 4 & -2.7e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{11.3}$$

$$B = \begin{bmatrix} 6.7e-3 & 0 \\ 7.7e-6 & 0 \\ -2.3e-5 & 0 \\ -3.8e-4 & 0 \\ 2.9e-4 & 0 \\ 4.9e-3 & 0 \\ -1.9e-3 & 0 \\ -1.6e-2 & 0 \\ 2.7e-3 & 0 \\ 7.3e-3 & 0 \\ 2.8e-3 & 0 \\ 2.0e-3 & 0 \\ -1 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -2.2 & -3.1 & -0.45 & -2.3 & -0.97 & -2.0 & -1.1 & -2.2 & -1.3 & -3.7 & -4.1 & -2.0 & 23.1 \end{bmatrix}$$

Similarly, the Fy LQR weightings are Q=50 and R=.075 for the error in Fy and the ufy control. The same estimator weightings are used as for the Fx LQE design, resulting in the following controller:

$$A = \begin{bmatrix} -18.2 & -11.6 & -1.9 & -5.2 & -3.2 & -11.3 & -12.2 & -15 & -13.0 & -11.5 & -8.6 & -19.1 & 103.3 \\ -10 & -173.4 & -231 & -72.0 & -125.3 & -137.4 & -168.4 & -146.2 & -67.0 & -53.6 & -30.7 & -14.9 & 0 \\ 0 & 1024 & 0 & 0 & 0 & -6.2e-9 & -5e-8 & -5.1e-7 & -1.9e-6 & -8.2e-6 & -9.7e-6 & -3.6e-5 & 0 \\ 0 & 0 & 512 & 0 & 0 & -5.5e-9 & -4.4e-8 & -4.5e-7 & -1.7e-6 & -7.2e-6 & -8.6e-6 & -3.2e-5 & 0 \\ 0 & 0 & 0 & 256 & 0 & 3.0e-7 & -2.4e-6 & 2.4e-5 & 9.0e-5 & 3.9e-4 & 4.7e-4 & 1.7e-3 & 0 \\ 0 & 0 & 0 & 0 & 128 & 2.0e-7 & 1.6e-6 & -1.7e-5 & 6.2e-5 & 2.7e-4 & 3.2e-4 & 1.2e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 128 & -1.2e-5 & -1.2e-4 & -4.4e-4 & -1.9e-3 & -2.3e-3 & -8.6e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.1e-6 & 128 & -9e-5 & -3.3e-4 & -1.5e-3 & -1.7e-3 & -6.5e-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.1e-6 & -5.7e-5 & 128 & 2.2e-3 & 9.4e-3 & 1.1e-2 & 4.2e-2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.8e-6 & -2.2e-5 & 2.3e-4 & 64 & 3.7e-3 & 4.4e-3 & 1.7e-2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -8.7e-6 & -7e-5 & -7.2e-4 & -2.7e-3 & 64 & -1.4e-2 & -5.2e-2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2e-6 & -1.6e-5 & -1.6e-4 & -6.1e-4 & -2.7e-3 & 16 & -1.2e-2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{11.4}$$

$$B = \begin{bmatrix} 4.6e-3 & 0 \\ 2.2e-6 & 0 \\ 1.9e-5 & 0 \\ 1.7e-5 & 0 \\ -9.0e-4 & 0 \\ -6.1e-4 & 0 \\ 4.4e-3 & 0 \\ 3.3e-3 & 0 \\ -2.2e-2 & 0 \\ -8.5e-3 & 0 \\ 2.6e-2 & 0 \\ 6.1e-3 & 0 \\ -1 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -2.1 & -2.9 & -0.49 & -1.3 & -0.80 & -2.8 & -3.1 & -3.8 & -3.2 & -2.9 & -2.2 & -4.8 & 25.8 \end{bmatrix}$$

The bandwidth plots for the LQG Fx and Fy controllers are shown in Figure 11.3 and Figure 11.4, respectively. The bandwidth for the Fx controller is about 4 rad/s with a peak of about 70 rad/s (near the 1st mode). Importantly, the bandwidth plot never is greater than 0 dB's, so no frequencies are amplified.

Similarly, the Fy controller bandwidth is also about 4 rad/s. The first 4 modes can be seen in the plot, but all are well suppressed.

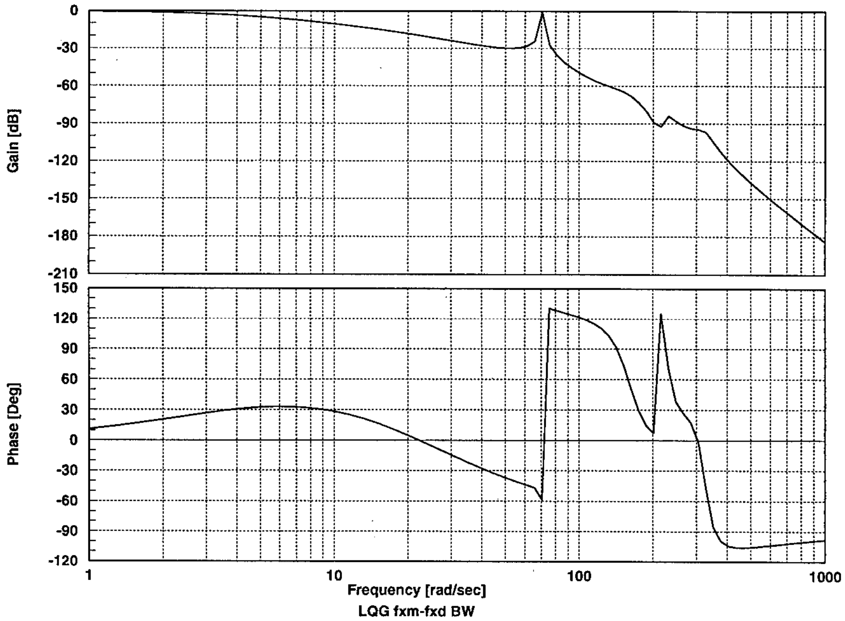


Figure 11.3- LQG Fx Controller Bandwidth

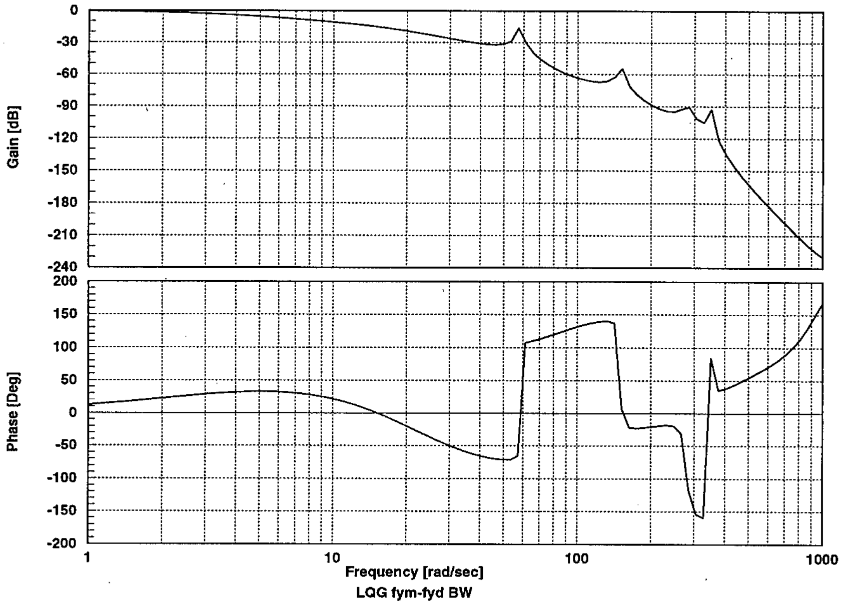


Figure 11.4- LQG Fy Controller Bandwidth

11.2.3 LQG Reduced-Order Force Control

The LQG controllers designed in the previous section are both 13th order controllers. The same two techniques applied to reduce the LQG position controllers are reapplied here to these LQG controllers. Namely, balanced truncation and modal residualization techniques are used. When applied to the LQG Fx controller, both balanced truncation and modal residualization produced 5th-order controllers. The balanced truncation controller was slightly more accurate than the modal residualization in matching the full-order controller by examination of the Bode plots.

The resulting 5th-order LQG balanced truncation controller is:

$$A = \begin{bmatrix} -3.5 & -63.4 & 6.0 & 2.0 & 8.1 \\ 63.4 & -0.41 & 31.1 & 0.55 & -2.7 \\ 6.0 & -31.1 & -10.6 & -3.8 & -13.1 \\ -2.0 & 0.55 & 3.8 & -2.1 & -2.1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -3.1e-4 & 0 \\ 2.6e-4 & 0 \\ -2.5e-3 & 0 \\ -6.4e-4 & 0 \\ -1 & 1 \end{bmatrix} \qquad (11.5)$$

$$C = \begin{bmatrix} -8.1 & -2.7 & 13.1 & 2.1 & 23.3 \end{bmatrix}, D = \begin{bmatrix} 2.8e-6 & 0 \end{bmatrix}$$

For the LQG Fy controller, both reduction techniques produced 4th-order controllers. Once again, examination of the Bode plot showed that the balanced truncation reduced-order controller was a slightly better representation of the full-order controller. Therefore, the balanced truncation controllers were used experimentally. The balanced truncation reduced-order LQG Fy controller is:

$$A = \begin{bmatrix} -2.5e-2 & -58.0 & -0.61 & -1.2 \\ 58.0 & -3.7e-2 & -1.56 & 1.4 \\ -0.61 & 1.56 & -19.7 & -19.7 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1.4e-3 & 0 \\ 2.2e-4 & 0 \\ -1.2e-3 & 0 \\ -1 & 1 \end{bmatrix} \qquad (11.6)$$

$$C = \begin{bmatrix} 1.2 & 1.4 & 19.7 & 28.1 \end{bmatrix}, D = \begin{bmatrix} -2.0e-4 & 0 \end{bmatrix}$$

11.2.4 SANDY Force Control Design

The SANDY controller is designed with a roll-off filter and an integral state for a second order controller. Essentially, this is a Proportional-Integral (PI) controller in series with a roll-off filter. There are two plants in the design - one for the command input and one for the noise input to the control. By allowing a plant with noise on the input, a penalty is placed on all frequencies and allows the optimization to consider the flexible modes. The weightings for the optimization are Q1=10 and R1=0 for the Fx command error, while the Fx noise error has weightings of Q2=1 and R2=.2. The optimization starts from a roll-off of 15/(s+15) and C matrix gains of 0.1 (corresponding to very low PI gains). The A matrix is not optimized to keep the filter and integral state eigenvalues the same, and the D matrix is not optimized to preserve the roll-off filter structure. Also, a constraint of -0.2 is placed on the maximum real part of the closed-loop eigenvalues. The resulting SANDY Fx controller is:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -15 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 10.8 \end{bmatrix}, C = \begin{bmatrix} 12.5 & 3.2 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad (11.7)$$

The bandwidth plot for the SANDY Fx controller is shown in Figure 11.5. Here, the controller is shown to have a bandwidth of about 12 rad/s with a peak around 70 rad/s, as with the LQG Fx controller.

In a similar manner, the SANDY controller is again optimized around a roll-off filter, this time at 25/(s+25). This design involves the equivalent two plants as the Fx design, but with new weightings of Q1=2.5, R1=0, Q2=1, and R2=.2. Again, the starting point for the optimization is the fixed roll-off filter and C matrix gains of 0.1.

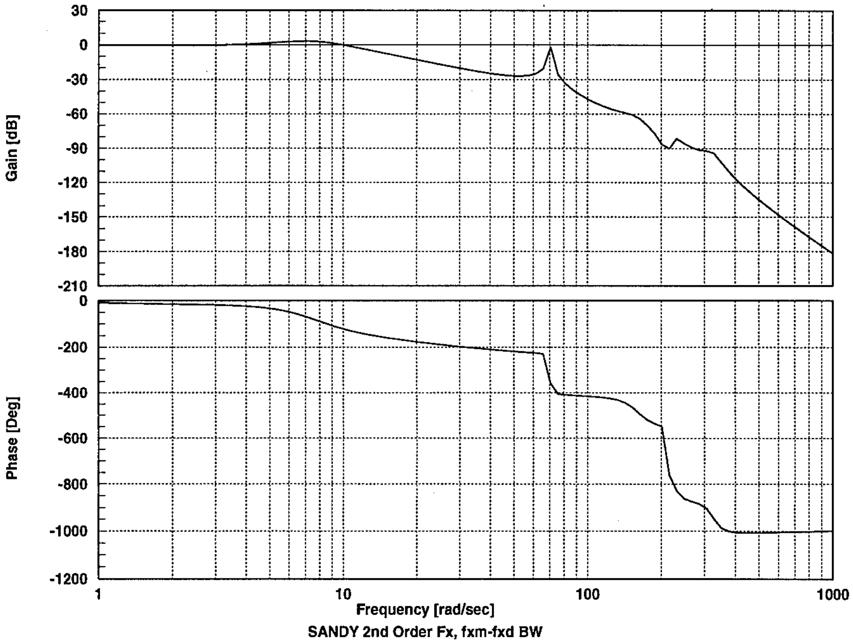


Figure 11.5- SANDY Fx Controller Bandwidth

Also, the same maximum closed-loop eigenvalue constraint of -.2 is applied to the optimization, and the same variables are optimized as before. The resulting SANDY Fy controller is:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -25 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 12.2 \end{bmatrix}, C = \begin{bmatrix} 14.3 & 4.0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad (11.8)$$

The corresponding bandwidth plot for the SANDY Fy controller is shown in Figure 11.6. In this case, the bandwidth is around 11 rad/s, and the first four modes are sufficiently attenuated.

11.3 Experimental Force Controller Comparison

The test profile for the force controllers is chosen as a step command from 25 to 50 gf and then back down to 25 gf. This tests the controllers for applying and removing force while maintaining contact. An LQG position controller is added to hold the arm at the same position, which is in contact with a straight line metal surface. For both the Fx and Fy tests, θ_1 is in the 0^o position and θ_2 is in the 90^o position. The metal surface is in the same orientation as the force sine sweep tests.

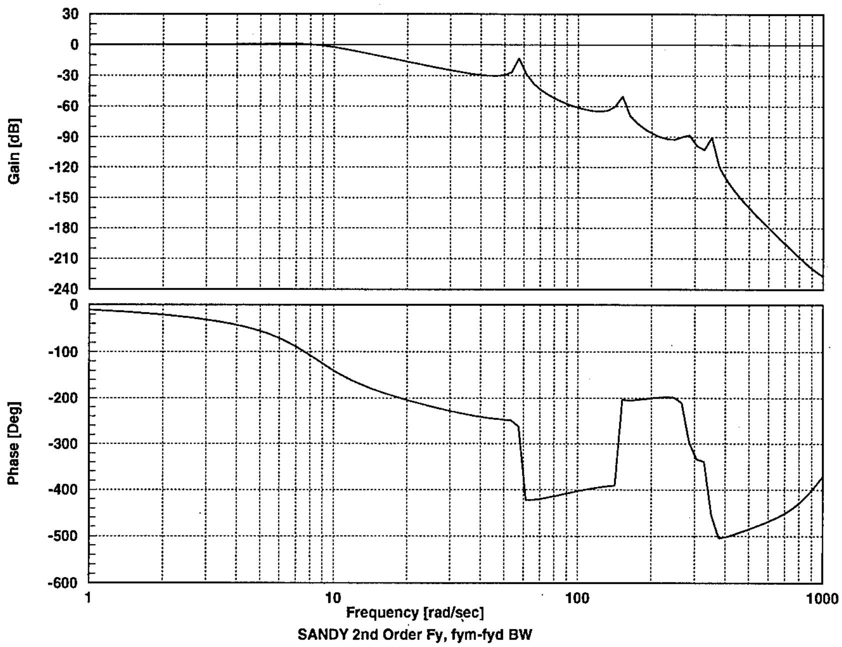


Figure 11.6- SANDY Fy Controller Bandwidth

Figure 11.7 shows the results for the Fx force controller tests. It is clear from the plot that the response of the LQG, reduced-order LQG, and SANDY Fx controllers have similar responses. The PI controller has a slower response time than all the others.

Likewise, the Fy force controller tests are shown in Figure 11.8. Once again, the LQG, reduced-order LQG, and SANDY controllers all have comparable performance. Also, the PI controller performance is again slower than the others.

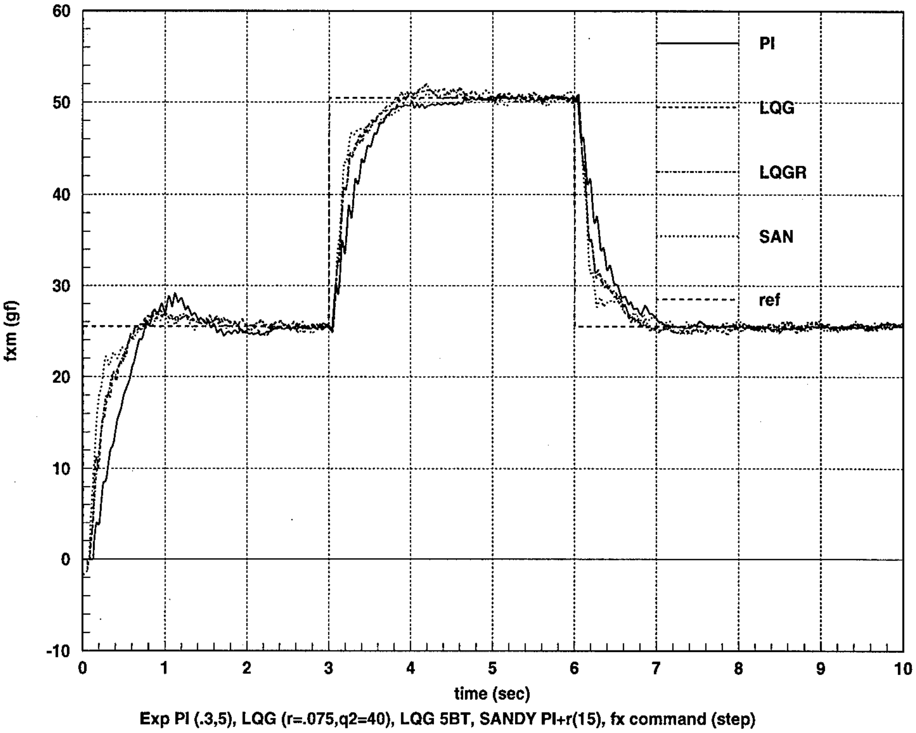


Figure 11.7- Experimental Controller Fx Comparison

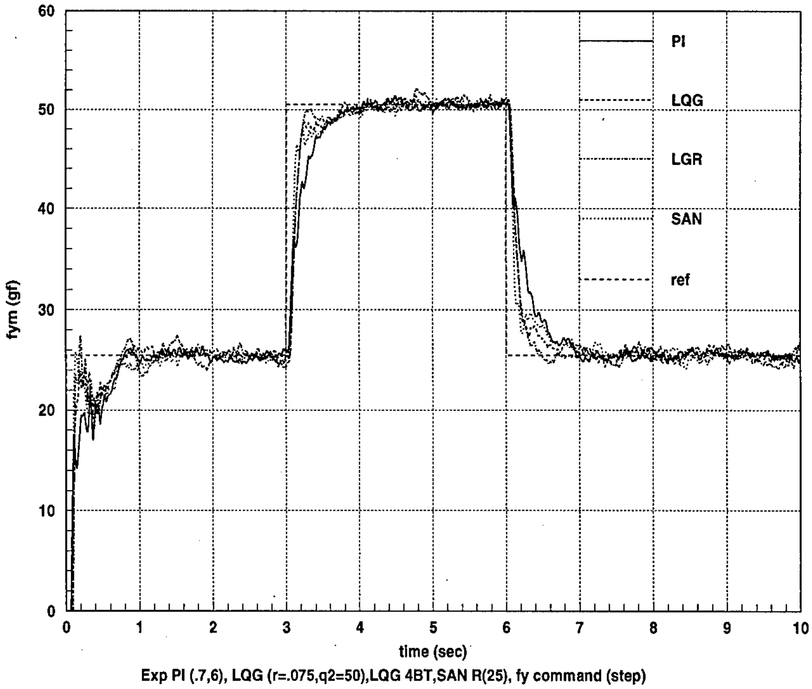


Figure 11.8- Experimental Controller Fy Comparison

12 Hybrid Control Designs

There are primarily two ways to design for a hybrid control scheme. The first is superposition, where the force controllers and the position controllers are designed independently and combined as a hybrid controller. The second method involves a complete hybrid optimization, where all controllers are designed simultaneously. The advantage of the superposition method is that it is simpler, and previous position and force controller designs can be used. However, it involves adjusting three separate controllers, and can be very time consuming. The complete optimization approach is more complicated since it involves larger plant models and controllers, but allows one to simultaneously affect all three controllers. Both methods can provide good results, depending on the amount of time spent on the design.

12.1 Superposition

The first cut uses the robust reduced-order SANDY position and reduced-order SANDY force controllers described in previous chapters. However, the position controller is not bandwidth separated from the force controllers, and has a tendency to jerk the arm off of the surface after the force control starts. The reason for the sudden surge in position control is that the force control initially opposes the position control while allowing a force buildup on the surface. Kraft's suggestion of bandwidth separation is used [34], and requires a new SANDY position controller design.

The new robust reduced-order controller is optimized around a fixed roll-off filter at 15 rad/s for the 1st input (θ_1 error) and another roll-off filter at 16 rad/s for the 2nd input (θ_2 error). The new weightings were Q1=5, Q2=22.5, and Q3=0.5,0.5 for each set of pseudo-plants corresponding to the 6 plant conditions (18 total pseudo plants). Q1 is the weighting on the θ_1 input, Q2 is the weighting on the θ_2 input, and Q3 is the weighting on noise inputs to both θ_1 and θ_2. The starting point of the optimization is a controller with similar performance to the original SANDY position controller. The resulting SANDY controller is 6th order, with 2 integral states, 2 roll-off states, and 2 dynamic states. This is a strongpoint of the SANDY program - it allows the designer to combine classical techniques, such as roll-off filters, with optimization techniques. The resulting SANDY controller has the form found in Equation (12.1) and Equation (12.2).

$$A = \begin{bmatrix} -15 & 0 & 0 & 0 & 0 & 0 \\ 0 & -16 & 0 & 0 & 0 & 0 \\ 3.75 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 217 & 184 & 0 & 0 & -19.7 & -104 \\ -32 & -22.5 & 0 & 0 & 2.1 & 6.9 \end{bmatrix}, B = \begin{bmatrix} 4 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{12.1}$$

$$C = \begin{bmatrix} -57 & -80 & -1.7 & 0.4 & 8.4 & 43 \\ 42 & 38 & -6 & 1.3 & -2.9 & -15 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{12.2}$$

The filtered bandwidth plots for the new robust 6th order SANDY controller are shown in Figure 12.1 and Figure 12.2. The bandwidth for the shoulder loop is about 2.75 rad/s while the bandwidth for the elbow loop is about 1.1 rad/s. Once again, a command prefilter at 1/(s+1) is added since that is how the controller was optimized and because the position reference is fed through a spline. The spline acts like a filter in the sense that the reference command is not instantaneously applied.

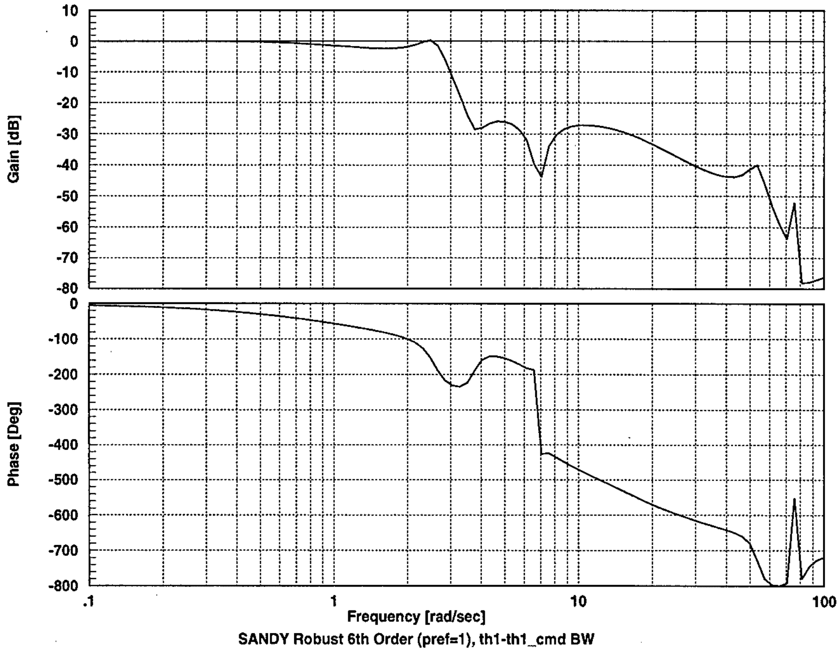


Figure 12.1- 6th Order Robust SANDY Shoulder Control Bandwidth

The new position controller is combined with the previous SANDY force controllers, and experimentally evaluated and compared to the full-order LQG position and force controllers. The overall SANDY hybrid controller is 10th order, and the overall LQG hybrid controller is 46th order.

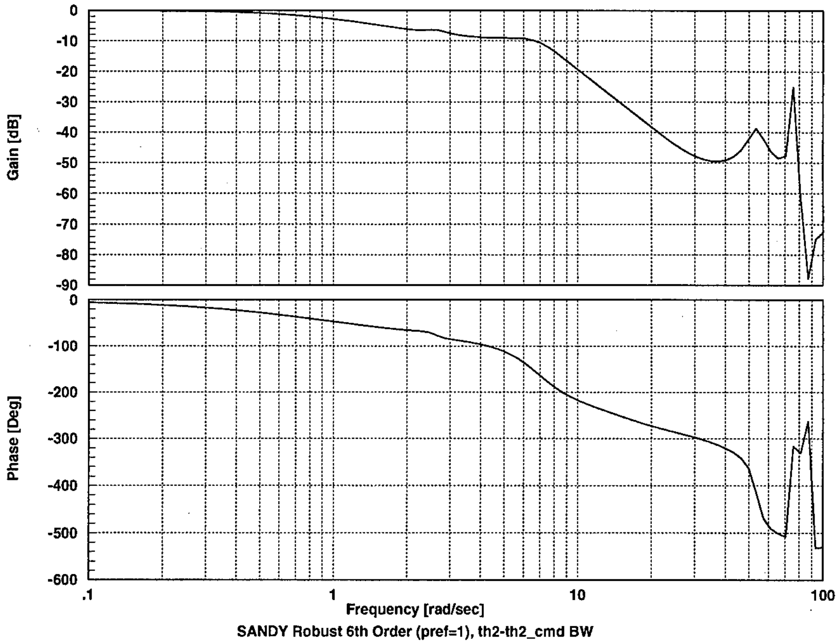


Figure 12.2- 6th Order Robust SANDY Elbow Control Bandwidth

In order to test the hybrid control scheme, the end-point is started in contact with a straight smooth metal surface. The exact trajectory is known and provides the position reference. In this case, a 0.4m line is traversed in 10 seconds for the position reference. Knowing the trajectory is not uncommon in manufacturing tasks when the shape of an item being machined is known but the user wants to apply a constant force (for example, to polish a surface). Thus, the task is to follow a straight line surface while applying a 50 gf magnitude surface normal force. This particular line is a challenge in the sense that it requires the Fx force to switch from positive to negative in the middle of the run.

To compare the controllers, first the full-order LQG position controller is used with the two LQG force controllers previously described. Next, the reduced-order SANDY position controller is used in conjunction with the three sets of force controllers (LQG, LQG reduced-order, and SANDY). All the experiments are presented in a strip chart showing, in order, the commanded voltage to the actuators, the position torques, the force torques, and the measured forces. Note that the measured magnitude is the resultant force of the measured x and y forces. Additionally, one plot of the end-point position is included with a time reference (marked by an ‘x’ at 0.5 sec intervals). The position plots are all similar since all combinations follow the same surface, so only one is included. The units for the voltage commands are Volts, for the torque commands are Newton-meters (Nm), and for the forces are grams-force (gf). When interpreting the strip charts, realize that the force control activation occurs when the control torques for the forces become non-zero.

Six hybrid control combinations are tested to compare reduced-order controller performance to full-order LQG controller performance using superposition of the force and position controllers. The combinations are listed in Table 12.1.

Table 12.1- Hybrid Superposition Combinations

Test Case	Position Controller	Force Controller
1	LQG (Full)	LQG (Full)
2	LQG (Full)	LQG (Reduced)
3	LQG (Full)	SANDY (2nd)
4	SANDY (6th)	LQG (Full)
5	SANDY (6th)	LQG (Reduced)
6	SANDY (6th)	SANDY (2nd)

The first test case matches the 20th order LQG position controller with the full-order (13th) LQG Fx and Fy force controllers. Figure 12.3 shows the end-point position during the test. Notice from the x every 0.5 seconds on the plot that the end-point starts out slow, speeds up during the middle part of the slew, and then slows. The “choppiness” is due to encoder resolution (about +/- 2mm).

The corresponding strip chart is seen in Figure 12.4. Notice that after the force control is activated at about 3.5 seconds, it takes about 1.5 seconds to initially reach the desired force and about another 1.5 seconds to settle after a 10 gf overshoot.

Next, the reduced-order LQG force controllers (Fx = 5th order and Fy = 4th order) are tested with the 20th order LQG position controller for the second test case.

Figure 12.5 shows that there is a slightly degraded performance from the full-order controller.

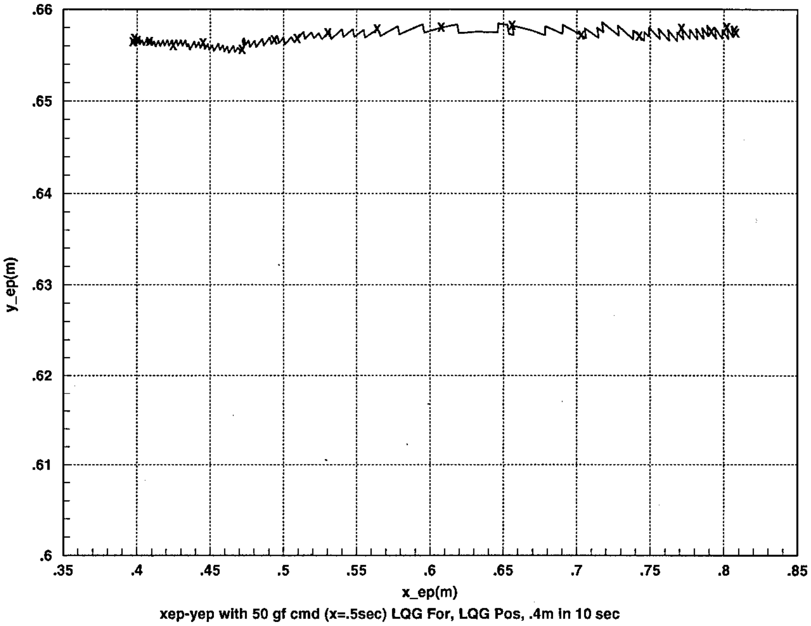


Figure 12.3- LQG Position and LQG Force End-Point History

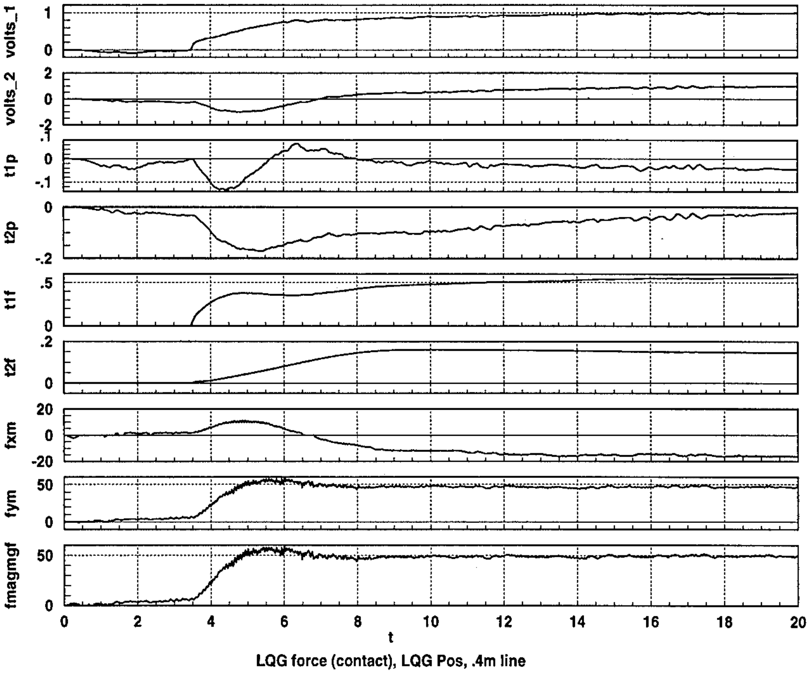


Figure 12.4- LQG Position and LQG Force Experimental Data

It takes about 1.75 seconds to initially reach the 50 gf point, and about 2 more seconds to settle. Also, there is slightly more overshoot for the reduced-order LQG force controllers.

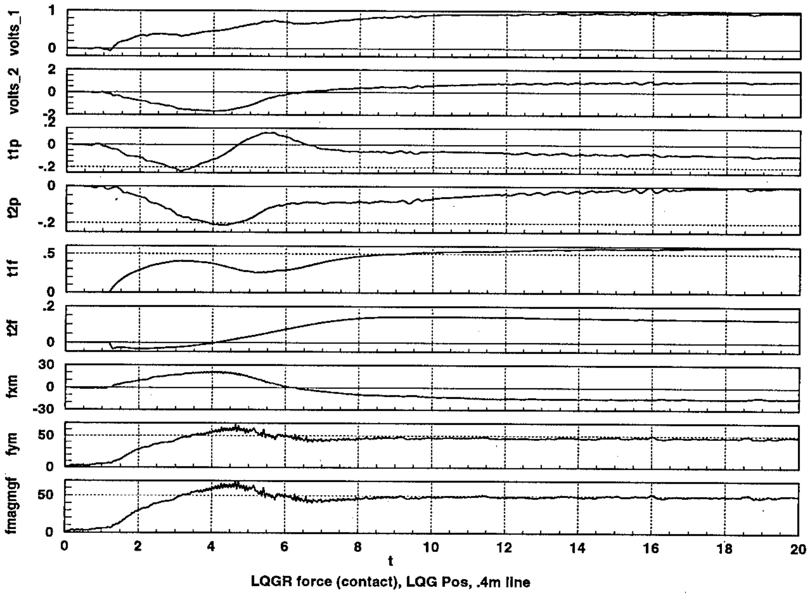


Figure 12.5- LQG Position and LQGR Force Experimental Data

The third test case matches the full-order LQG position controller with the reduced-order SANDY force controllers (Fx =2nd order and Fy = 2nd order). Figure 12.6 shows that it takes about 1.25 seconds to build up to the required 50 gf with no overshoot. This is actually slightly better than the LQG force controller combination!

The next phase of experiments uses reduced-order SANDY position controllers. The first force controllers used with the SANDY position controller are the LQG full-order (13th) controllers. Figure 12.7 shows the resulting experimental run for test case 4. This time, after initial activation, it takes roughly 1.5 seconds to reach the 50 gf level, and then roughly 1.5 additional seconds to stabilize after an undershoot. The initial undershoot is about 20 gf, which is more than the LQG controller. However, the response times are about the same.

The same trends continue for the force controllers with the SANDY position controller during the 5th test case. Figure 12.8 shows that the reduced-order LQG force controllers show slight degradation from full-order LQG force controllers in the form of slightly more overshoot. The initial response time is about 2 seconds, and it takes an additional 3 seconds to stabilize. The overshoot is about 15 gf in this case.

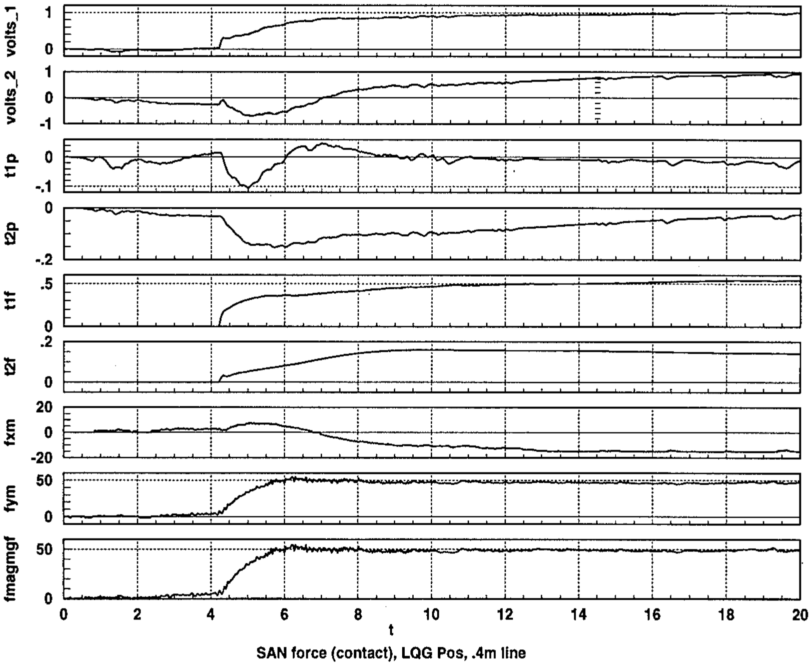


Figure 12.6- LQG Position and SANDY Force Experimental Data

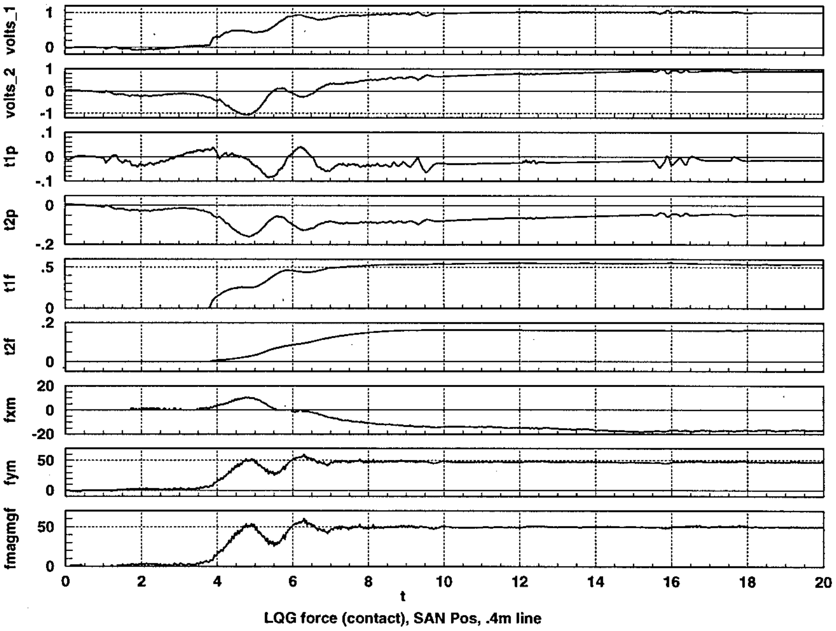


Figure 12.7- SANDY Position and LQG Force Experimental Data

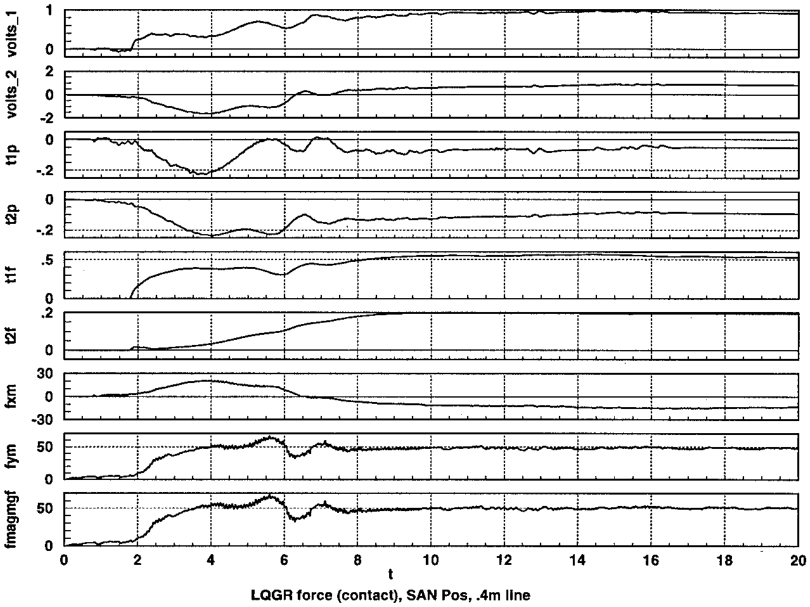


Figure 12.8- SANDY Position and LQGR Force Experimental Data

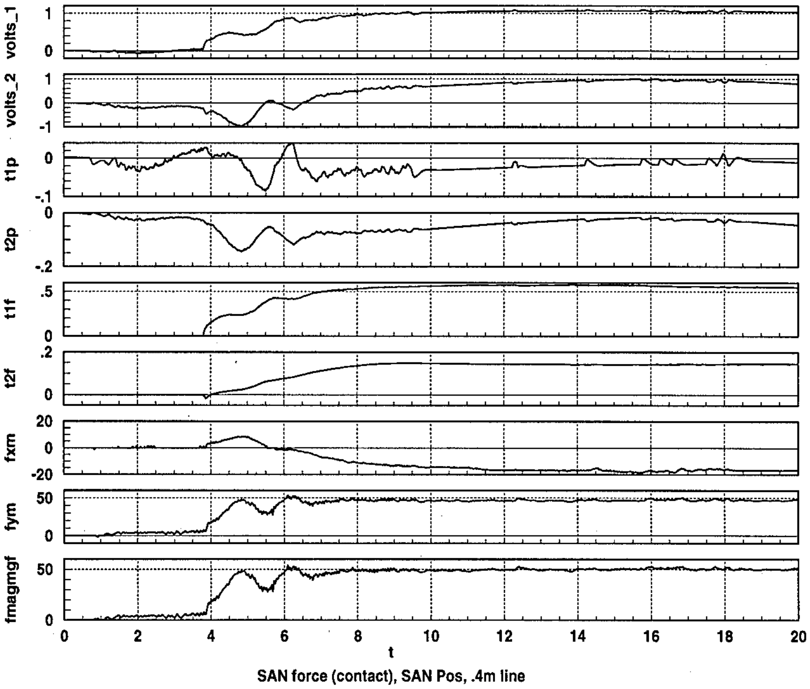


Figure 12.9- SANDY Position and Force Experimental Data

Finally, Figure 12.9 shows the SANDY position and force experimental data for the 6th test case. There is once again a slight improvement (less overshoot) over the

LQG force controllers. However, compared to the LQG position control, there is a little more overshoot with about the same rise time.

Overall, the SANDY reduced-order force controllers provide slightly better performance than the full-order LQG force controllers. The LQG position controller provides slightly better performance than the SANDY reduced-order position controller. Thus, for hybrid control, the reduced-order SANDY controllers provide comparable performance to the full-order LQG controllers.

12.2 Complete Optimization

The complete optimization can involve strictly a reduced-order design using one plant, or can involve a robust reduced-order design. Both approaches are covered in this section. In both cases, the hybrid controllers have the structure shown in Equation (12.3) and Equation (12.4).

$$A_{hyb} = \begin{bmatrix} A_p & 0 & 0 \\ 0 & A_{fx} & 0 \\ 0 & 0 & A_{fy} \end{bmatrix}, B_{hyb} = \begin{bmatrix} B_p & 0 & 0 \\ 0 & B_{fx} & 0 \\ 0 & 0 & B_{fy} \end{bmatrix} \tag{12.3}$$

$$C_{hyb} = \begin{bmatrix} C_p & 0 & 0 \\ 0 & C_{fx} & 0 \\ 0 & 0 & C_{fy} \end{bmatrix}, D_{hyb} = \begin{bmatrix} D_p & 0 & 0 \\ 0 & D_{fx} & 0 \\ 0 & 0 & D_{fy} \end{bmatrix} \tag{12.4}$$

In these equations, matrices with the subscript p correspond to the position controller. Also, the fx and fy subscripts correspond to the Fx and Fy controllers. From this structure, the position inputs are fed only to the position controller while the fx and fy inputs are fed only to their respective controllers. This allows the controllers to perform just like their superposition counterparts while allowing simultaneous or complete optimization within SANDY.

12.2.1 Complete Reduced-Order Optimization

To start with, a reduced-order hybrid control design is used. This involves one position model, one Fx model, and one Fy model combined into one hybrid position and force control model. The same pseudo-plants used for the previous position and force control designs are used, which results in 3 pseudo-plants for position, 2 for Fx, and 2 for Fy. Thus, one hybrid plant requires 7 pseudo-plants for the hybrid optimization, as shown in Table 12.2.

Table 12.2 - Hybrid Pseudo-Plants

Pseudo-Plant	Purpose
1	Optimize for θ_1 error
2	Optimize for θ_2 error
3	Optimize for high freq (noise into both θ control inputs)
4	Optimize for Fx errors
5	Noise into Fx control input
6	Optimize for Fy errors
7	Noise into Fy control input

The actual hybrid plant combines the position, Fx, and Fy plants in parallel, so the new plant is 46th order and has 4 control inputs, 4 reference inputs, and 6 outputs. The control and reference inputs are for θ_1, θ_2, fx, and fy. The outputs are, in order, $\dot{\theta}_1$, $\dot{\theta}_2$, θ_1_error, θ_2_error, fx_error, and fy_error. The $\dot{\theta}$ terms are used only in the criterion, thus the resulting SANDY hybrid controller has 4 inputs (the error terms) and 4 outputs (the control inputs to the hybrid plant). This hybrid controller is actually a position controller, an Fx controller, and an Fy controller combined in parallel. Thus, the only coupling between the controllers is in the optimization cost function. Functionally, the complete hybrid SANDY controller has the same structure as the superposition method. This particular design uses a position model linearized around the θ_1=0° and θ_2=90° nominal conditions. The force models were also obtained at the same angle positions.

The optimization weightings are shown in Table 12.3. The 4 control weightings apply to τ_{1p}, τ_{2p}, u_{fx}, and u_{fy}. The actual implementation applies the Jacobian to the force commands to generate τ_{1f} and τ_{2f}. The position and torque forces are added together to produce the actual τ_1 and τ_2 applied to the plant. In addition, the maximum real eigenvalue is constrained to -0.2 while adding a minimum damping constraint of 0.7 for all frequencies below 25 rad/s. This damping constraint allows the rigid body modes to have a minimum damping while not constraining the flexible modes. Without additional sensors and/or actuators, the damping on these flexible modes can't be effectively controlled. The starting point for the optimization is the previous superposition hybrid controller. In this way, none of the previous optimization work is wasted.

Table 12.3 - Hybrid Optimization Weightings

Pseudo-Plant	Criterion Penalty	Control Penalty
1	Q1=diag(0,0,5,0,0,0)	R1=diag(0,0,0,0)
2	Q2=diag(0,0,0,22.5,0,0)	R2=diag(0,0,0,0)
3	Q3=diag(0,0,0.2,0.2,0,0)	R3=diag(0,0,0,0)
4	Q4=diag(0,0,0,0,10,0)	R4=diag(0,0,0,0)
5	Q5=diag(0,0,0,0,1,0)	R5=diag(0,0,.2,0)
6	Q6=diag(0,0,0,0,0,2.5)	R6=diag(0,0,0,0)
7	Q7=diag(0,0,0,0,0,1)	R7=diag(0,0,0,.2)

These weightings produce the complete hybrid controller shown in Equations (12.5) through (12.8).

$$\boldsymbol{A}_p = \begin{bmatrix} -15 & 0 & 0\ 0 & 0 & 0 \\ 0 & -16 & 0\ 0 & 0 & 0 \\ 3.75 & 0 & 0\ 0 & 0 & 0 \\ 0 & 4 & 0\ 0 & 0 & 0 \\ 201.6 & 186.1 & 0\ 0 & -17.1 & -97.9 \\ -12.2 & -24.7 & 0\ 0 & 2.3 & 10.9 \end{bmatrix}, \boldsymbol{B}_p = \begin{bmatrix} 4 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{12.5}$$

$$\boldsymbol{C}_p = \begin{bmatrix} -26.4 & -68.1 & -0.12 & 0.16 & 6.4 & 31.2 \\ 31.1 & 13.0 & -1.9 & 0.6 & -0.59 & -6.8 \end{bmatrix}, \boldsymbol{D}_p = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{12.6}$$

$$\boldsymbol{A}_{fx} = \begin{bmatrix} 0 & 1 \\ 0 & -15 \end{bmatrix}, \boldsymbol{B}_{fx} = \begin{bmatrix} 0 \\ 9.05 \end{bmatrix}, \boldsymbol{C}_{fx} = \begin{bmatrix} 11.6 & 3.3 \end{bmatrix}, \boldsymbol{D}_{fx} = \begin{bmatrix} 0 \end{bmatrix} \tag{12.7}$$

$$\boldsymbol{A}_{fy} = \begin{bmatrix} 0 & 1 \\ 0 & -25 \end{bmatrix}, \boldsymbol{B}_{fy} = \begin{bmatrix} 0 \\ 10.1 \end{bmatrix}, \boldsymbol{C}_{fy} = \begin{bmatrix} 15.6 & 2.6 \end{bmatrix}, \boldsymbol{D}_{fy} = \begin{bmatrix} 0 \end{bmatrix} \tag{12.8}$$

The position controller has the same inputs and outputs as the superposition position controllers. Namely, the inputs are error_θ_1 and error_θ_2 while the outputs are τ_{1p} and τ_{2p}. The bandwidth plot of the shoulder motor in Figure 12.10 shows that the bandwidth is about 1.75 rad/s. The elbow motor bandwidth shown in Figure 12.11 reveals a bandwidth of about 1.2 rad/s.

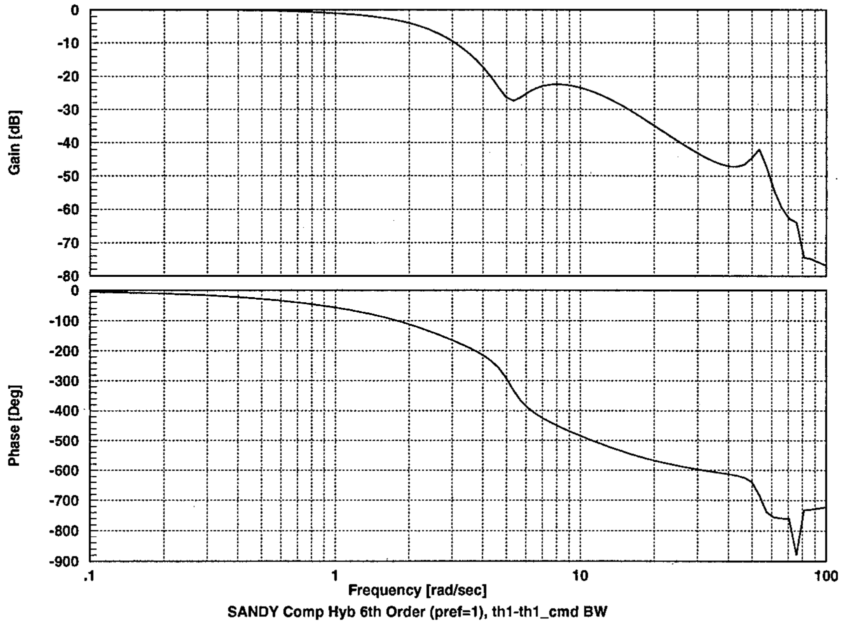


Figure 12.10- Shoulder Bandwidth, SANDY Hybrid Position Controller

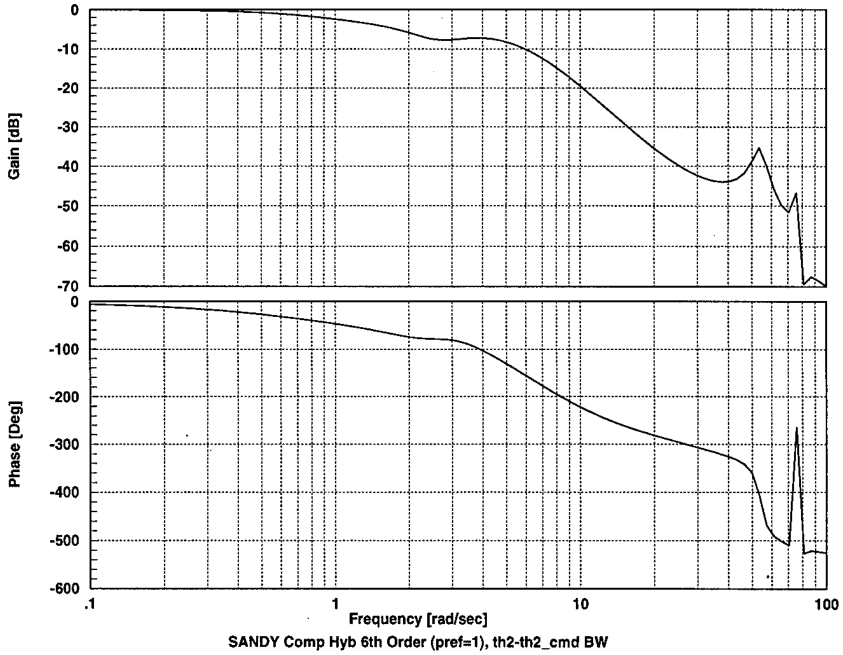


Figure 12.11- Elbow Bandwidth, SANDY Hybrid Position Controller

The force bandwidth for the Fx controller is shown in Figure 12.12 as 11 rad/s while the Fy bandwidth shown in Figure 12.13 is 8 rad/s.

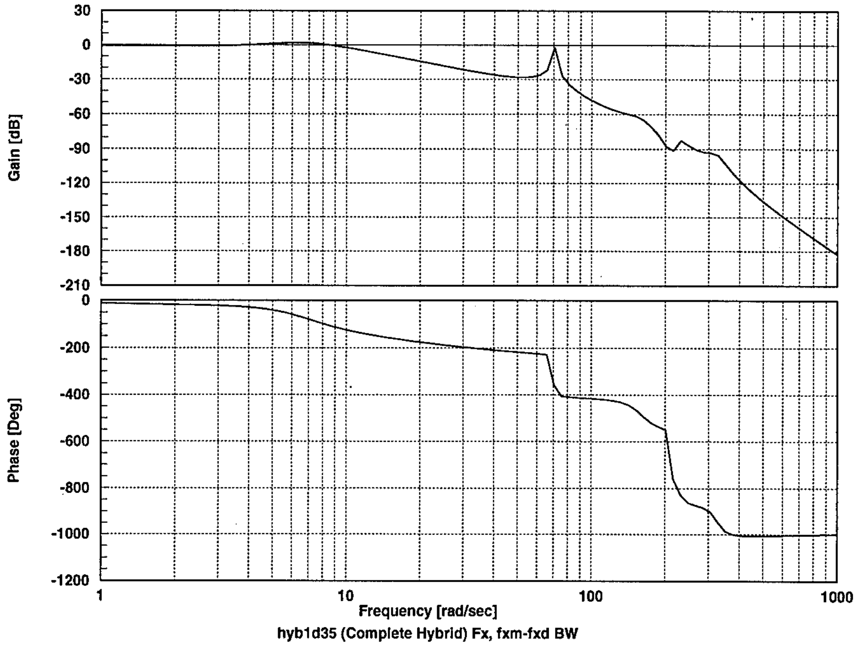


Figure 12.12- Fx Control Bandwidth, SANDY Hybrid Controller

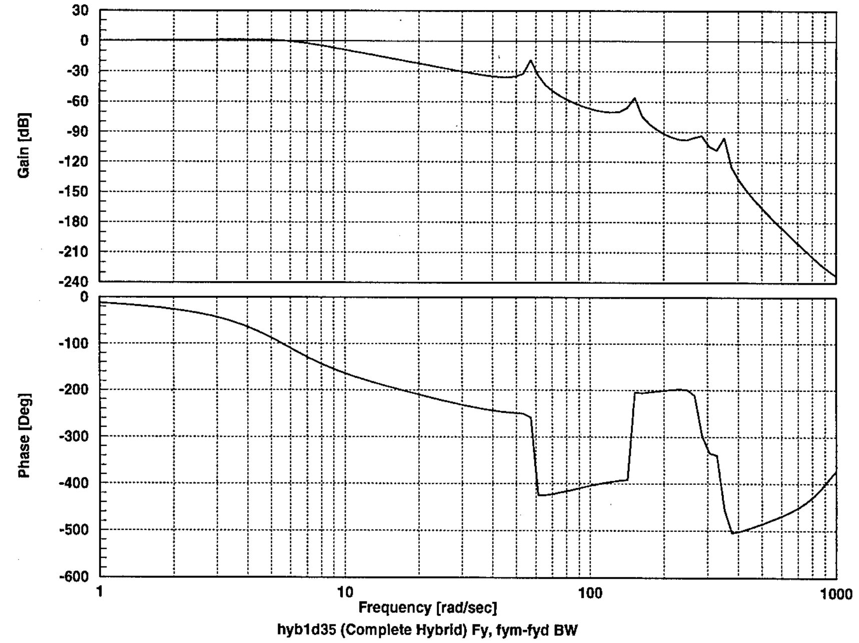
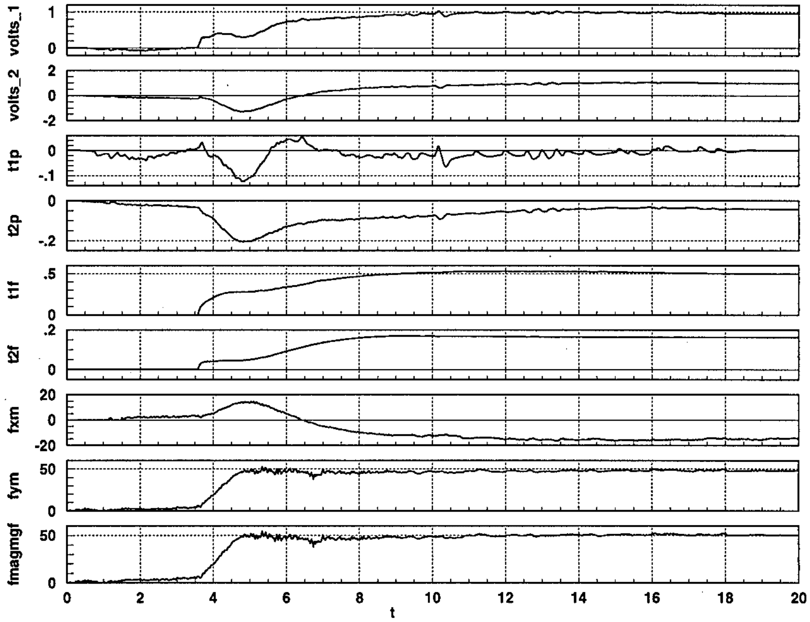


Figure 12.13- Fy Control Bandwidth, SANDY Hybrid Controller

The resulting complete hybrid controller is tested along the same trajectory as before, and the results are shown in Figure 12.14. It takes slightly over 1 second to build up to 50 gf with no overshoot, and thus gives slightly better performance than the LQG position and force controller.



hyb1d35 (.7 min - 25) SANDY force (contact), SANDY Pos, .4m line

Figure 12.14- Complete Hybrid Optimization Experimental Data

12.2.2 Complete Robust Reduced-Order Optimization

The next logical step is to add multiple plant conditions to allow a robust reduced-order hybrid design. This particular effort uses 2 position models linearized at different angle positions. Logical extensions for robustness could involve variable tip mass and even more linearized angle position models. Additionally, multiple force models could be added. The cost of adding more plants is to increase the required computation time in the optimization. In addition, there is a classic trade-off between robustness and performance. Thus, one should only include the conditions which the plant will frequently operate in to avoid unnecessary performance degradation in the primary area of operation.

The first position plant is linearized about the θ_1=20^o and θ_2=110^o point, while the second position plant is linearized about the θ_1=-20^o and θ_2=70^o point. The key variation here is the θ_2 angle, which varies by +/- 20^o from the plant in the original design. The same force models are used in the hybrid plant, which combines the position model, the Fx force model, and the Fy force model.

The same optimization constraints are applied to this design as the original complete hybrid optimization, namely real eigenvalue and frequency dependent damping constraints. The new weightings are listed in Table 12.3. Notice that the penalties had to be increased on the position controller to maintain acceptable performance.

Table 12.4 - Robust Hybrid Optimization Weightings

Pseudo-Plant	Criterion Penalty	Control Penalty
1	Q1=diag(0,0,10,0,0,0)	R1=diag(0.01,0,0,0)
2	Q2=diag(0,0,0,45,0,0)	R2=diag(0,0.01,0,0)
3	Q3=diag(0,0,0.4,0.4,0,0)	R3=diag(0.01,0.01,0,0)
4	Q4=diag(0,0,0,0,10,0)	R4=diag(0,0,0,0)
5	Q5=diag(0,0,0,0,1,0)	R5=diag(0,0,0.05,0)
6	Q6=diag(0,0,0,0,0,2.5)	R6=diag(0,0,0,0)
7	Q7=diag(0,0,0,0,0,1)	R7=diag(0,0,0,0.05)

These weightings produce the complete robust hybrid controller shown in Equations (12.9) through (12.12).

$$\boldsymbol{A_p} = \begin{bmatrix} -15 & 0 & 0 & 0 & 0 & 0 \\ 0 & -16 & 0 & 0 & 0 & 0 \\ 3.75 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 209.6 & 176.2 & 0 & 0 & -16.9 & -104.6 \\ -13.5 & -21.9 & 0 & 0 & 2.3 & 10.5 \end{bmatrix}, \boldsymbol{B_p} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{12.9}$$

$$\boldsymbol{C_p} = \begin{bmatrix} -26.9 & -64.1 & 0.68 & -0.21 & 6.0 & 29.4 \\ 28.1 & 9.4 & -1.0 & 0.5 & -0.48 & -6.9 \end{bmatrix}, \boldsymbol{D_p} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{12.10}$$

$$\boldsymbol{A_{fx}} = \begin{bmatrix} 0 & 1 \\ 0 & -15 \end{bmatrix}, \boldsymbol{B_{fx}} = \begin{bmatrix} 0 \\ 9.52 \end{bmatrix}, \boldsymbol{C_{fx}} = \begin{bmatrix} 10.9 & 3.2 \end{bmatrix}, \boldsymbol{D_{fx}} = \begin{bmatrix} 0 \end{bmatrix} \tag{12.11}$$

$$\boldsymbol{A_{fy}} = \begin{bmatrix} 0 & 1 \\ 0 & -25 \end{bmatrix}, \boldsymbol{B_{fy}} = \begin{bmatrix} 0 \\ 10.3 \end{bmatrix}, \boldsymbol{C_{fy}} = \begin{bmatrix} 12.9 & 2.6 \end{bmatrix}, \boldsymbol{D_{fy}} = \begin{bmatrix} 0 \end{bmatrix} \tag{12.12}$$

The bandwidth plots for robust hybrid position shown in Figure 12.15 and Figure 12.16 show a shoulder bandwidth of 1.9 rad/s and an elbow bandwidth of 1.25 rad/s.

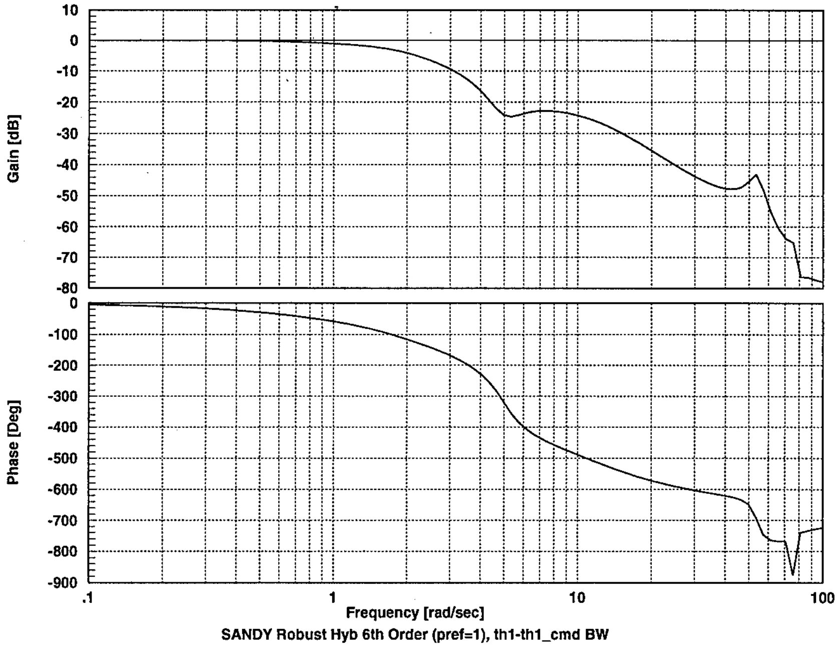


Figure 12.15- Shoulder Bandwidth, SANDY Robust Hybrid Controller

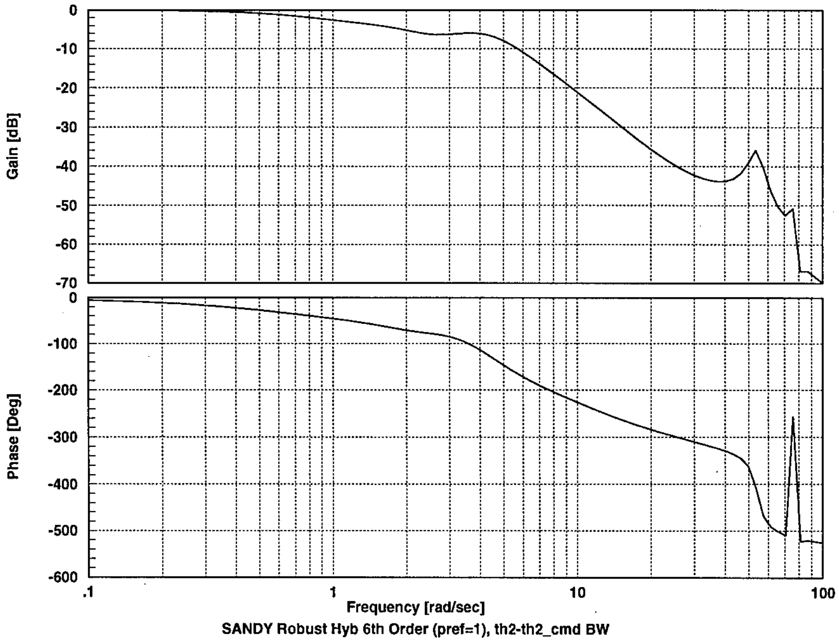


Figure 12.16- Elbow Bandwidth, SANDY Robust Hybrid Controller

The Fx bandwidth plot of Figure 12.17 shows a bandwidth of 11 rad/s while the Fy bandwidth plot of Figure 12.18 reveals a bandwidth of 8 rad/s.

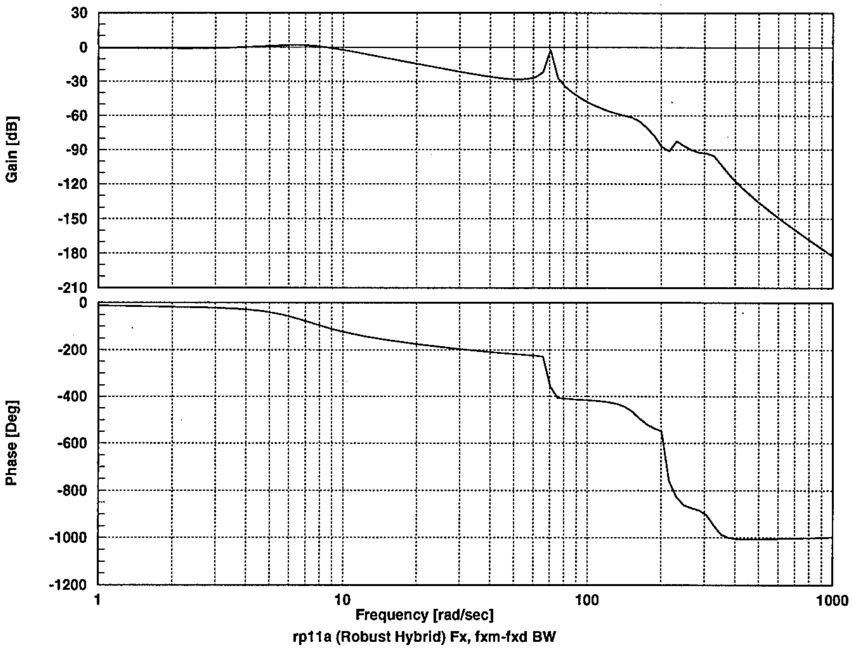


Figure 12.17- Fx Bandwidth, SANDY Robust Hybrid Controller

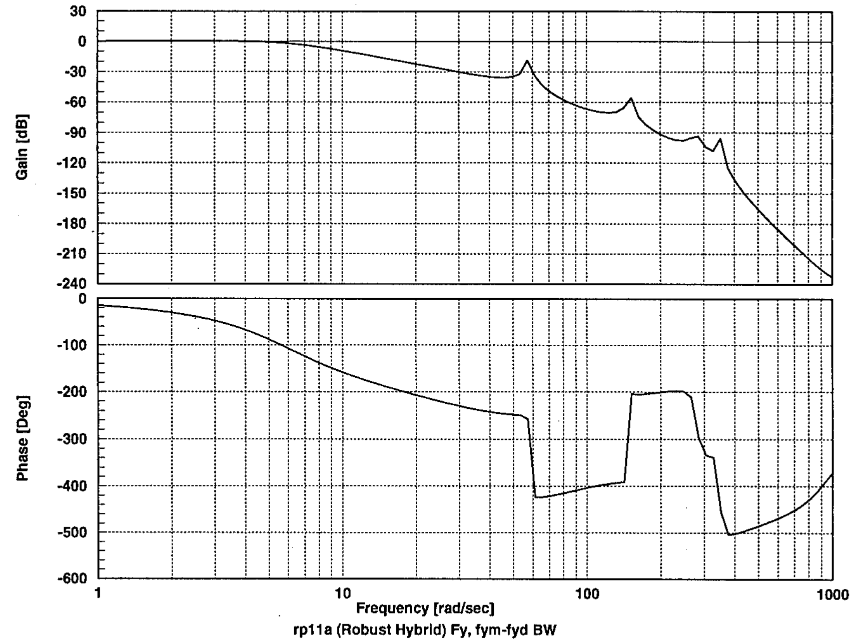
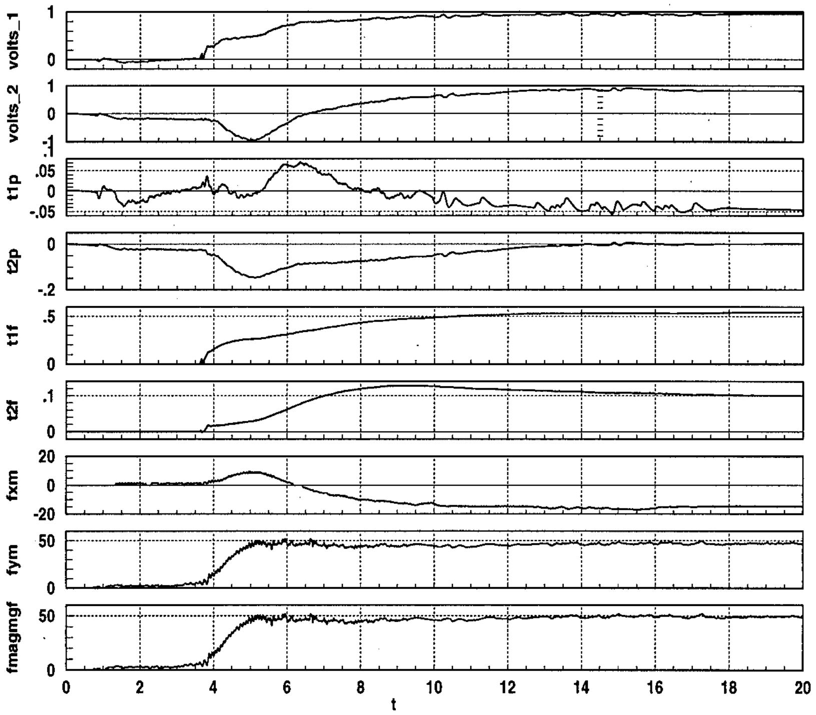


Figure 12.18- Fy Bandwidth, SANDY Robust Hybrid Controller

Figure 12.19 shows the experimental response of the robust reduced-order hybrid complete optimization controller. In this case, the response takes slightly over 1 second to ramp up to 50 gf and has no overshoot. Once again, this response is slightly better than the full-order hybrid LQG controller.



rp11a (.7 min - 25) SANDY force (contact), SANDY Pos, .4m line

Figure 12.19- Complete Robust Hybrid Optimization Experimental Data

12.3 Hybrid Design Experimental Conclusions

Overall, the reduced-order SANDY controllers perform comparably to the full-order LQG controllers. In addition, performing a complete hybrid optimization allowed better design results since it was easier to adjust all the controllers simultaneously. It is possible that a better design for superposition could be obtained by iteratively tuning the position, Fx, and Fy controllers. However, because three controllers are involved, it is more difficult than the complete optimization method

13 Surface Following Experiments

As noted earlier, there are many applications where a robotic manipulator is required to follow a surface and apply a constant force to perform a task. These applications include buffing, grinding, polishing, or painting. The normal approach to tackle these problems is to learn the desired trajectory and then perform the task. The obvious problem here is that every new shape requires learning a new trajectory.

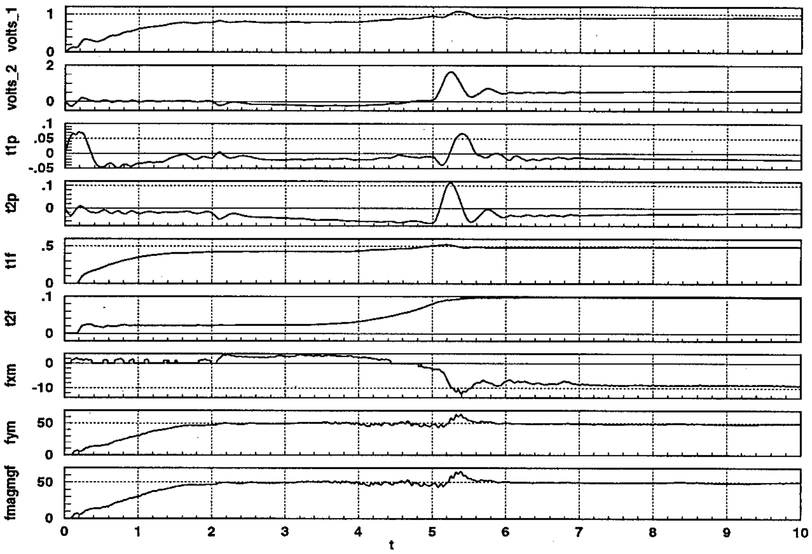
The particular application here is TLA. The sensors used are a wrist-mounted force sensor and optical encoders at the two hub angles. The tool mounted to the wrist is a low-friction roller which allows measurement of almost purely surface normal forces. Using the surface normal and hub angle information, a position command is given along the surface tangent. Thus, the algorithm allows movement along an unknown surface by continually commanding motion along the surface tangent. In order to see how effective the algorithm is for the TLA, different shapes are used. Also, the effect of different controllers and different commanded speeds is examined.

All the tests start with the end-point in contact with the surface, and then a slight position offset is commanded into the surface for 2 seconds to allow the force control to engage. Therefore, if the algorithm runs for 5 seconds, the arm actually moves for 3 seconds. This allows the force to build up to the desired level, and allows for a common starting state for all the experiments. Two plots are provided for each test - a strip chart which shows the measured forces and commanded torques, and an x-y end-point plot which shows the actual end-point location (calculated through forward kinematics) and the position reference provided by the surface following algorithm.

The strip chart units are volts for the physical voltage output to the power supplies, Newton-meters (Nm) for the torques, and gram-force (gf) for all the measured forces. The x-y plot has units of meters, and the dashed line is the reference output. There is an x to represent data at 0.5 second intervals on the x-y plot as well.

13.1 Surface Shape Effects

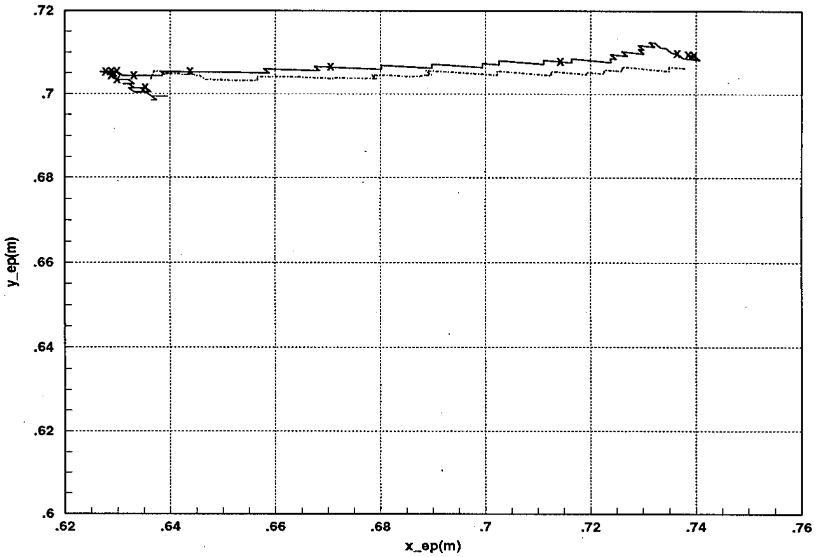
Many shapes are made up of lines and arcs, so it seems logical to test the algorithm on these shapes. The simplest surface is a straight line, and the first experiment commands 50 gf with a velocity of 1 cm/s while using the LQG hybrid controller for 5 seconds. The strip chart in Figure 13.1 shows that the force is maintained at 50 gf with a slight jump as the algorithm ends due to the sudden slowing of the arm.



LQG force, LQG Pos, unknown line, 1 cm per sec

Figure 13.1- LQG, Line, 50gf Command, 1 cm/s, Strip

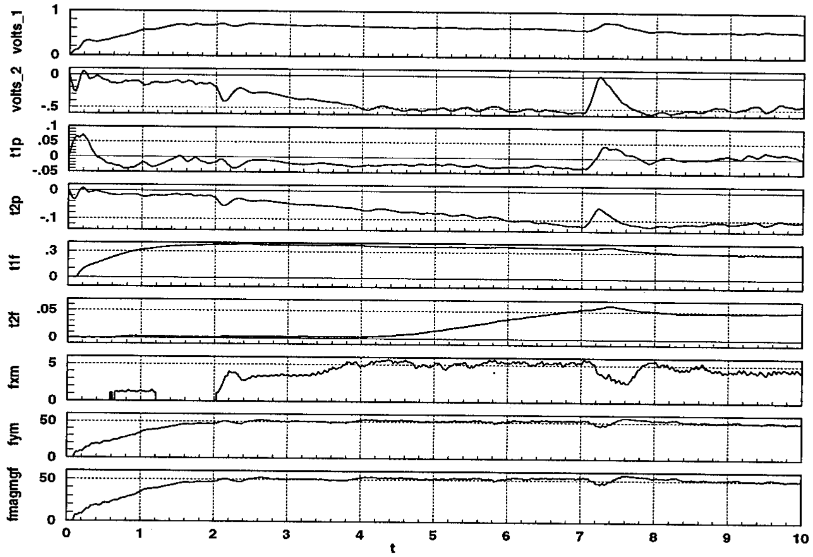
The x-y plot shown in Figure 13.2 shows that reference for a line is indeed a line. Thus, the algorithm successfully tracks a straight line surface.



xep-yep with 50 gf cmd (x=.5sec) LQG For and Pos, Unknown Line 1 cm per sec

Figure 13.2- LQG, Line, 50gf Command, 1 cm/s, x-y

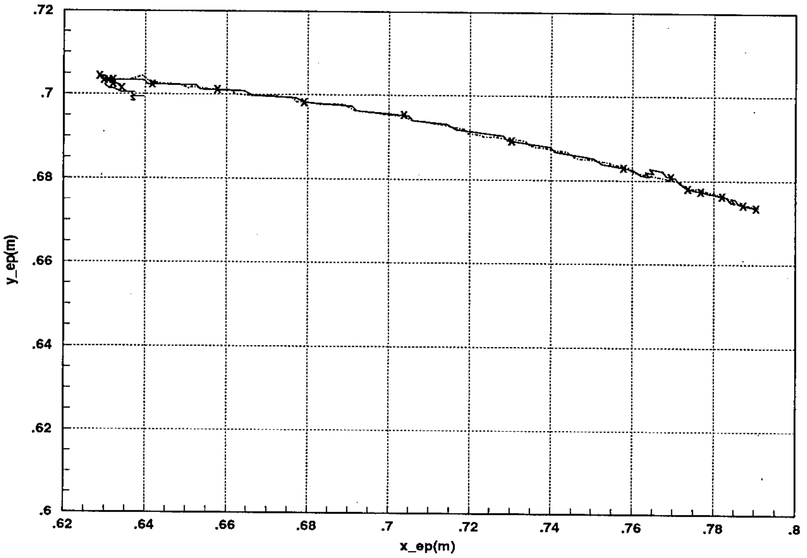
The next shape tested is a concave arc, which is another common surface encountered in manufacturing. The strip chart of Figure 13.3 shows that the force level is maintained at 50 gf during tracking while having only a slight transition while the arm is stopped after 7 seconds.



LQG for-pos, unk arc 50gf 1 cm per sec, 7 sec

Figure 13.3- LQG, Concave Arc, 50gf, 1 cm/s, Strip

Figure 13.4 shows that the reference and actual trajectory are both clearly a concave arc. In this case, the algorithm clearly works for a concave arc.



xep-yep with 50 gf cmd (x=.5sec) LQG For-Pos, Unk Arc 1 cm per sec, 7 sec

Figure 13.4- LQG, Concave Arc, 50gf, 1 cm/s, x-y

Another common surface encountered in manufacturing is a convex arc. In this case, the time for the test is 6 seconds while maintaining a 50 gf contact. The strip chart is shown in Figure 13.5, and shows that the force is maintained for the tracking, but that a large force overshoot is required to stop the arm after 6 seconds. This is because the arm tends to accelerate over the convex surface, even though the reference remains relatively smooth.

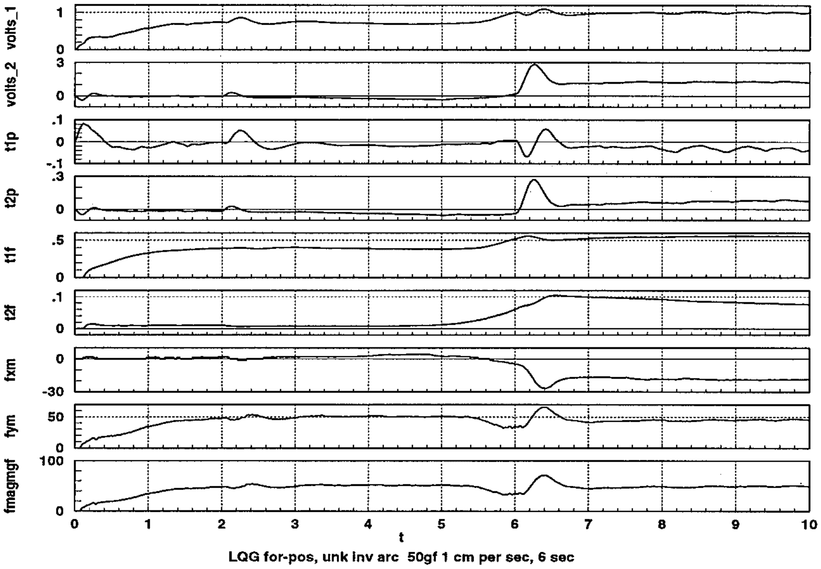


Figure 13.5- LQG, Convex Arc, 50 gf, 1 cm/s, Strip

The corresponding x-y plot in Figure 13.6 shows that the surface following algorithm successfully commands a convex arc trajectory. Notice the large overshoot in position due to the rapid acceleration over the convex arc.

Overall, the algorithm successfully provides a surface following position reference for all the shapes. The speed at which the arm moves is dependent on shape, and does not provide a direct correlation to the speed reference.

13.2 Effect of Tracking Speed Reference

The major way a user may affect the speed at which the algorithm tracks is by varying the value k in Chapter 10. Since the sampling time is constant, varying k changes the tracking velocity reference. As stated earlier, this reference velocity doesn't provide a direct measure of the velocity.

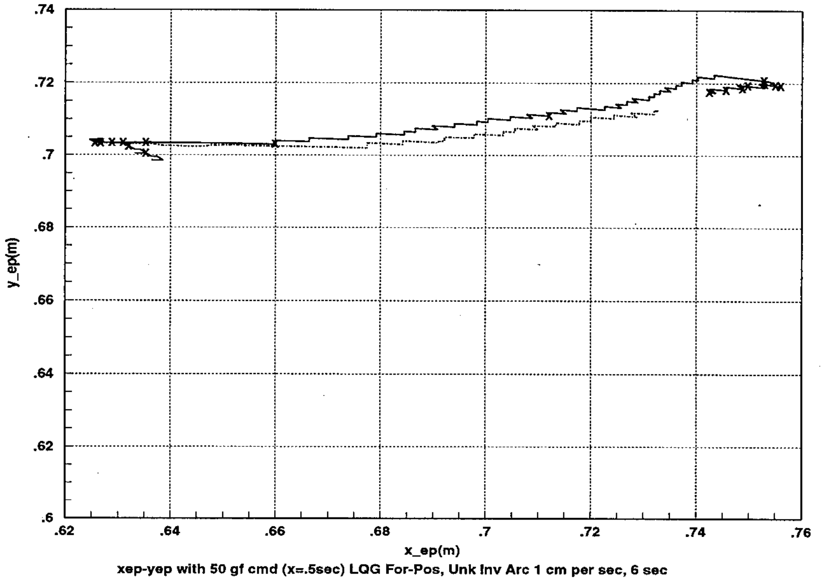


Figure 13.6- LQG, Convex Arc, 50 gf, 1 cm/s, x-y

Repeating the line tracking experiment for a 50 gf command and a 2 cm/s velocity rèference results in the strip chart shown in Figure 13.7. In this case, the algorithm is only run for 4 seconds due to the increased distance traveled. Notice that the force tracking has degraded slightly from the 1 cm/s velocity reference experiment for the line with slightly more force overshoot at the beginning and end of the trajectory.

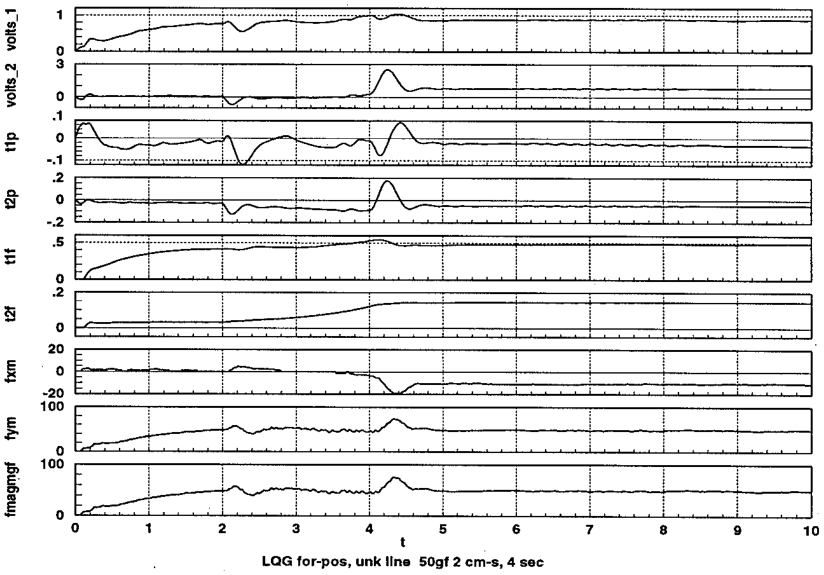


Figure 13.7- LQG, Line, 50gf Command, 2 cm/s, Strip

The corresponding x-y plot is shown in Figure 13.8. The algorithm once again provides a good line reference for the unknown surface it follows. Notice that the distance traversed is further in less time than the 1 cm/s velocity reference.

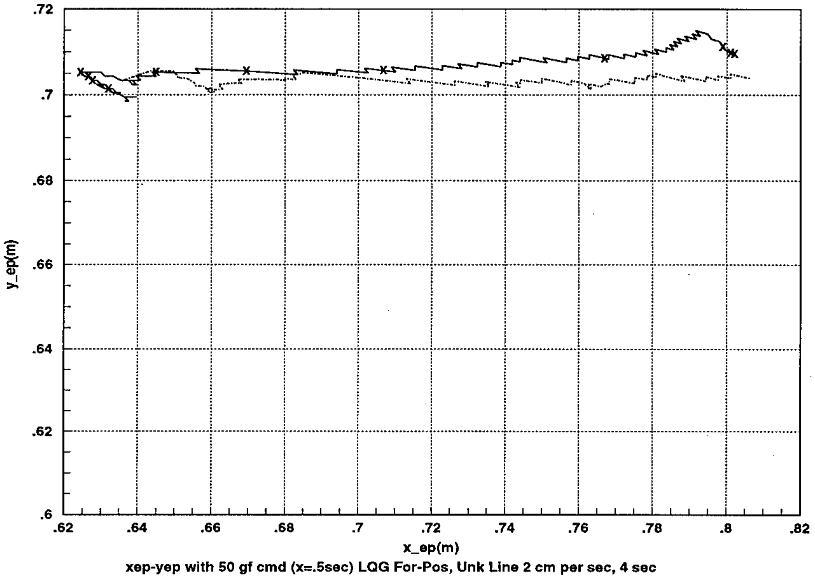


Figure 13.8- LQG, Line, 50gf Command, 2 cm/s, x-y

13.3 Effect of Commanded Force

All of the previous tests are conducted with a commanded contact force level of 50 gf. To ensure the algorithm will work at other force levels, a test at a 30 gf is performed.

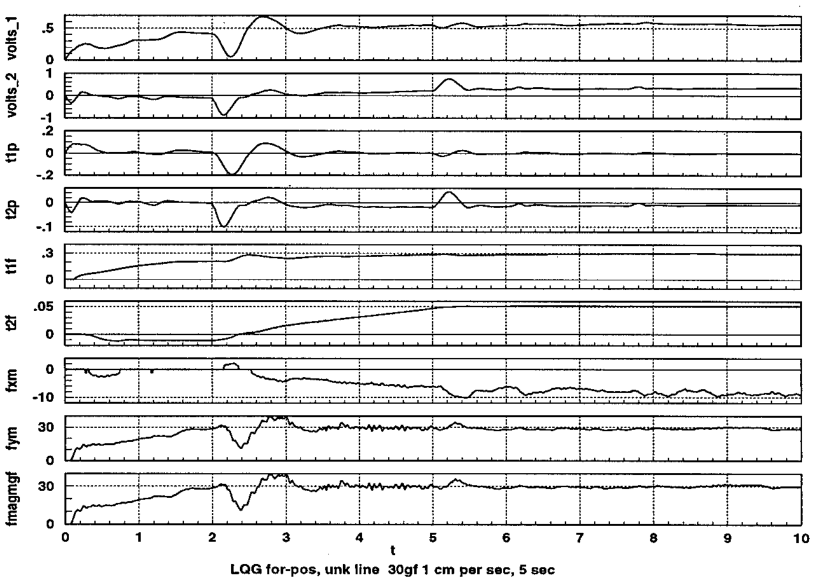
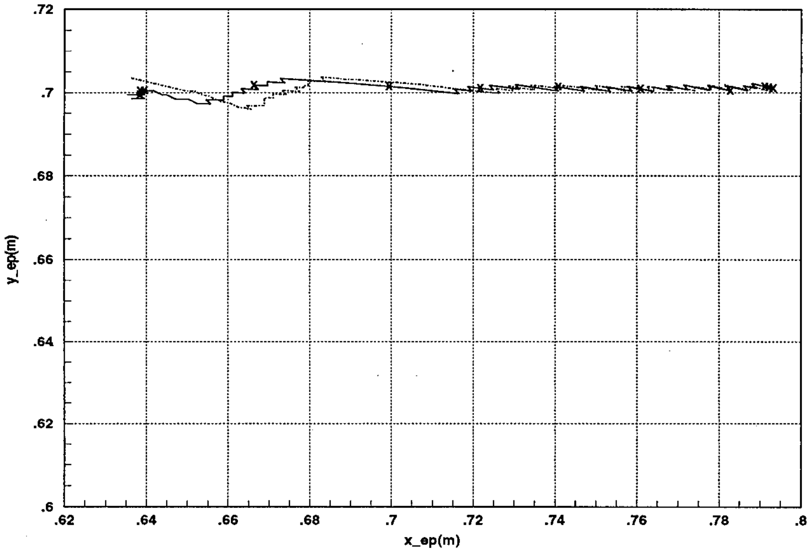


Figure 13.9- LQG, Line, 30gf Command, 1 cm/s, Strip

The same straight line surface is used with the LQG hybrid controller for another

5 second test run. The strip chart shown in Figure 13.9 reveals an initial transient in the force and then good tracking for the remaining time.

The corresponding x-y plot in Figure 13.10 shows after the initial transient that a straight line is well-represented by the algorithm. Also, the lower force allows the arm to start its movement more quickly due to the lower contact force with the wall. Once the movement is started, the tracking speed is roughly the same as the 50gf experiment.



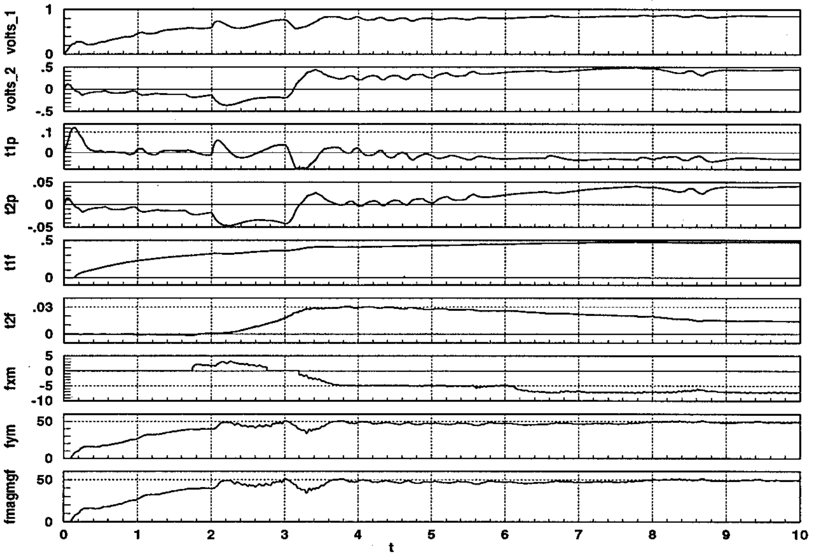
xep-yep with 30 gf cmd (x=.5sec) LQG For-Pos, Unk Line 1 cm per sec, 5 sec

Figure 13.10- LQG, Line, 30gf Command, 1 cm/s, x-y

13.4 Effect of Different Controllers

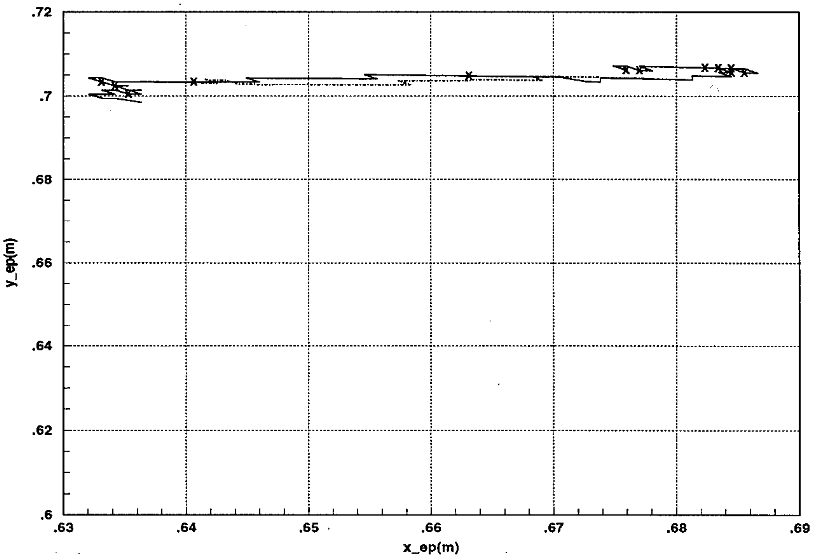
To verify that the algorithm isn't tuned to a full-order LQG controller, a reduced-order SANDY controller is also used. The SANDY controller has a higher bandwidth position controller, so it provides different performance than the LQG controller. As a basis of comparison, the straight line surface is used with a 50 gf command for 3 seconds. The resulting strip chart in Figure 13.11 shows that the force is maintained fairly well during tracking, with an undershoot in force at the end. It is interesting that this controller has an undershoot in force at the end of movement but the LQG controller has an overshoot.

Figure 13.12 clearly shows that a line reference is provided, even with a different controller. Notice that there is a faster speed for this higher bandwidth controller, which causes more overshoot and slightly degraded performance.



SAN for-pos, unk line 50gf 1 cm per sec, 3 sec

Figure 13.11- SANDY, Line, 50gf Command, 1 cm/s, Strip



xep-yep with 50 gf cmd (x=.5sec) SAN For-Pos, Unk Line 1 cm per sec, 3 sec

Figure 13.12- SANDY, Line, 50gf Command, 1 cm/s, x-y

13.5 Modified Tangent Speed Command

One thing the experiments show is that the speed at which the algorithm tracks varies with shapes, commanded force, and the bandwidth of the position controller. In addition, the arm speed seemed to increase as the algorithm progressed due to the projection of the last command on the tangent line becoming larger.

The modified algorithm commands a constant offset along the tangent line, instead of an offset plus a previous command projection. The expected effect is to provide a more uniform rate of tracking, although the startup time will be longer since the reference error will not increase. Using the illustration of the algorithm in Chapter 10, essentially the offset is commanded from point P_2 and not P_3.

First, the algorithm is applied to the same straight line as before with 50 gf applied for 5 seconds. Figure 13.13 shows in the resulting strip chart that the force is maintained at the 50 gf level during the tracking and at the end. Since the arm is moving more slowly, there is no force overshoot at the end.

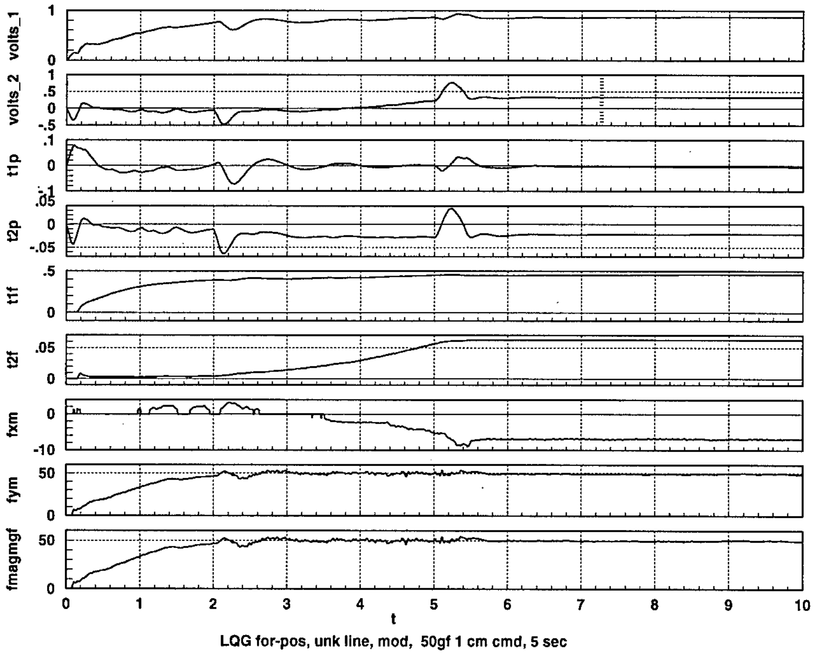


Figure 13.13- LQG, Line, 50gf, Mod, 1 cm cmd, strip

The corresponding x-y plot is shown in Figure 13.14. In this case, the distance traveled is slightly less than the original algorithm and the acceleration a little smoother. Clearly, the tracking aspect is unaffected by the change in speed in this case as the line reference is given for the unknown line surface.

Applying the modified algorithm to a concave arc for 15 seconds with 50 gf applied results in the strip chart shown in Figure 13.15. The resulting force profile is smooth with just a small force disturbance at the end of the run.

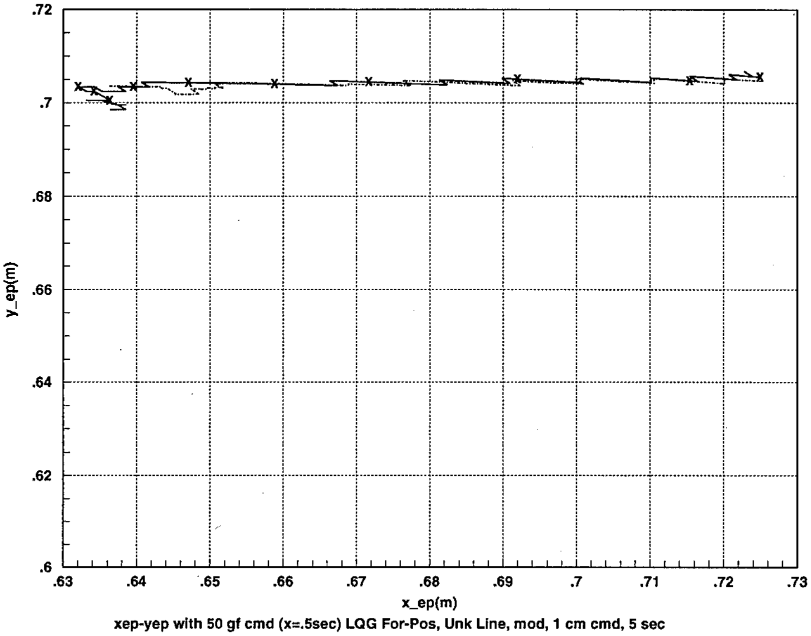


Figure 13.14- LQG, Line, 50gf, Mod, 1 cm cmd, x-y

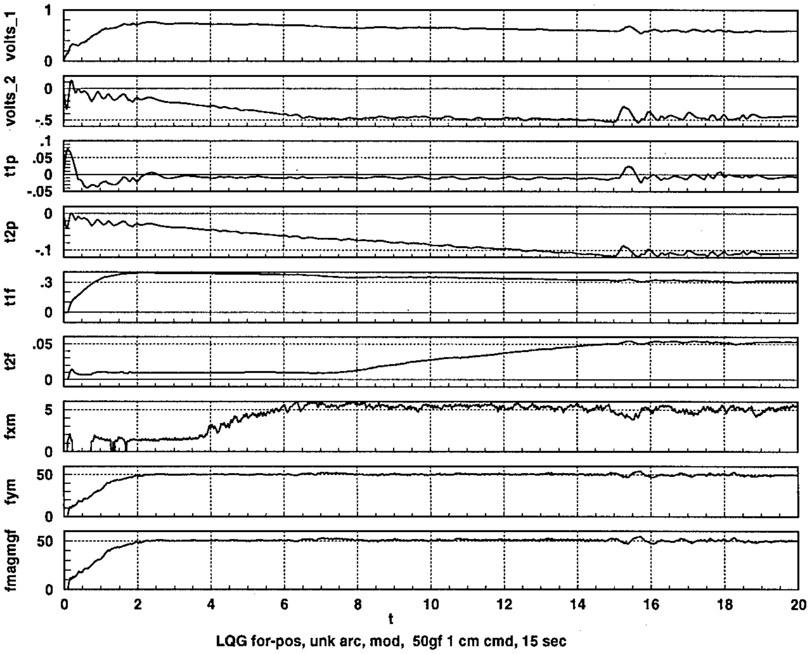


Figure 13.15- LQG, Concave Arc, 50gf, Mod, 1 cm cmd, strip

The x-y plot in Figure 13.16 shows a uniform spacing of the distance traveled per 0.5 second time hash. Thus, the modified algorithm regulates the speed better for this case. Notice, however, that it takes a lot longer for the algorithm to start since the position reference signal is constant.

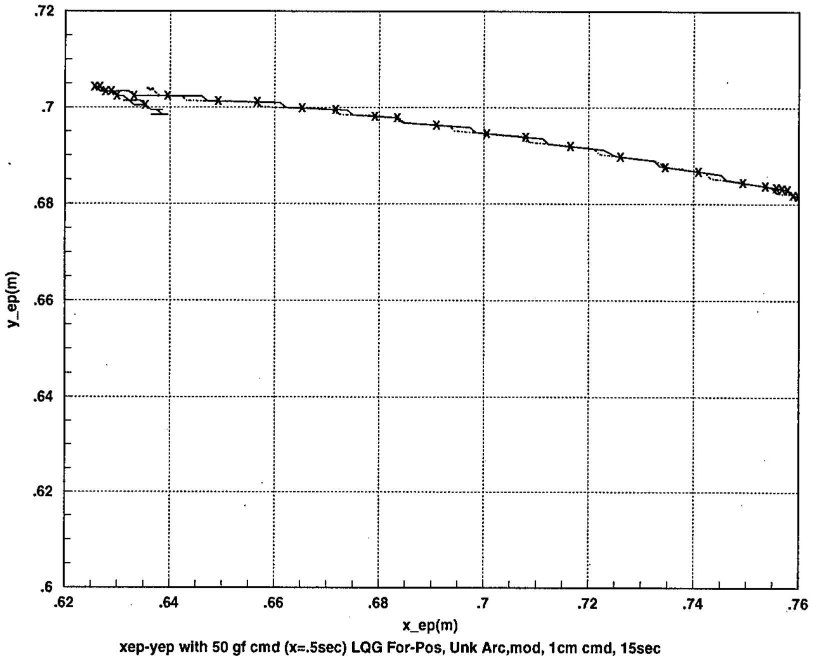


Figure 13.16- LQG, Concave Arc, 50gf, Mod, 1 cm cmd, x-y

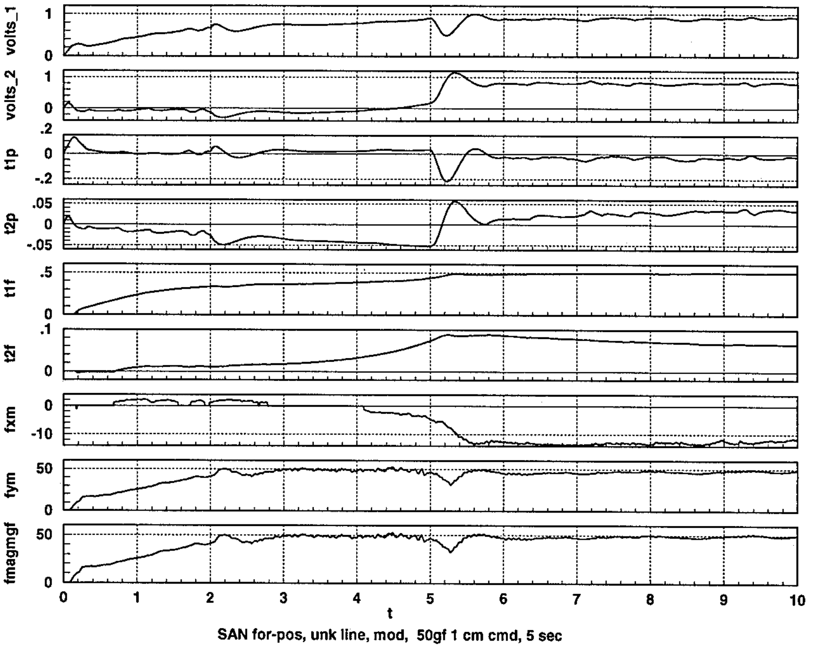


Figure 13.17- SANDY, Line, 50gf, Mod, 1 cm cmd, strip

Examining the effect of a different controller on the modified algorithm provides a good test. The SANDY controller is once again used as the basis of comparison for the straight line trajectory for 5 seconds while applying a 50 gf force. The strip chart shown in Figure 13.17 reveals that the force is well-maintained until the end of the run, when a force undershoot occurs.

Figure 13.18 shows that the distance traveled by the arm is further for the SANDY controller, and that there is still some acceleration as the run progresses. The higher speed causes a position overshoot and the force discrepancy at the end of the 5 second algorithm application. This shows that the algorithm's speed is still dependent on the position controller.

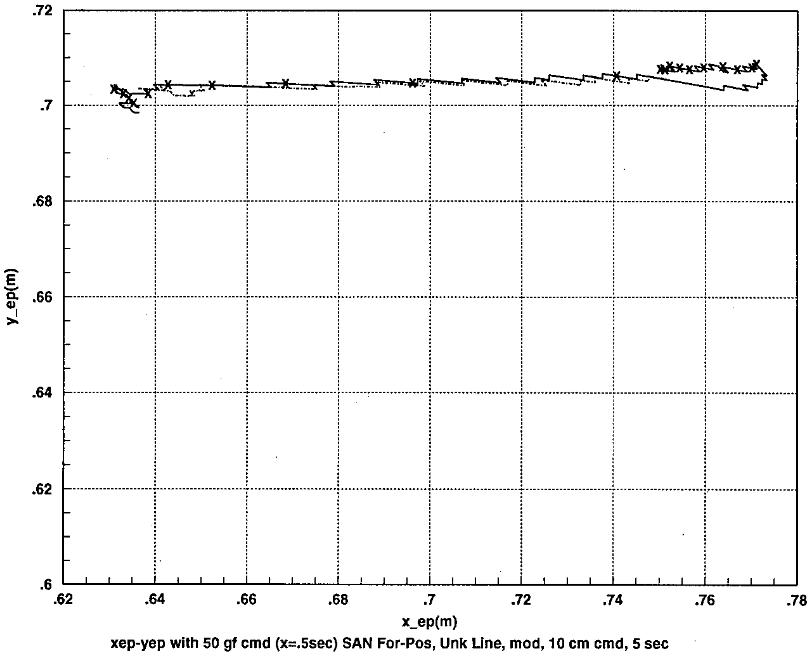


Figure 13.18- SANDY, Line, 50gf, Mod, 1 cm cmd, x-y

13.6 Surface Following Conclusions

The surface following algorithm works for all the shapes tested. In addition, different force levels, different commanded speeds, and different controllers all work successfully with the algorithm. However, the speed at which the algorithm tracks varies with shape, force level, and controller.

To remedy the speed variation concern, a modified surface following algorithm which commands a constant distance along the surface tangent was examined. The modified algorithm did a better job of providing constant speed, but there was still some acceleration once the arm was in motion. The speed increase was especially noticeable with the higher bandwidth SANDY controller. The disadvantage of the modified algorithm is that it takes longer to start, and it doesn't always overcome the initial stiction of the roller force contact with the surface.

14 Surface Touchdown

One of the more challenging aspects of robotics is performing tasks while in contact with a surface. In such a contact case, not only the position but the force must be controlled. This is called a constrained motion problem if the surface is rigid, or a "compliant motion task" for a compliant surface [52]. However, the transition from free-space motion to post-contact force control can be even more challenging. This surface touchdown problem must address the tendency of the robot to "bounce" after hitting the surface. Once the surface contact is lost, the force control is no longer valid. Thus, a successful touchdown will avoid any bounce. This touchdown phase of the robotic task is sometimes called impact control [47].

Kraft's baseline hybrid control scheme is described in Chapter 10. When used for touchdown, the block diagram must be modified to allow pure position control when not in contact with a surface. Thus, a block is added which checks the force level, as shown in Figure 14.1. When the measured force is less than the defined threshold, the force control output is set to zero.

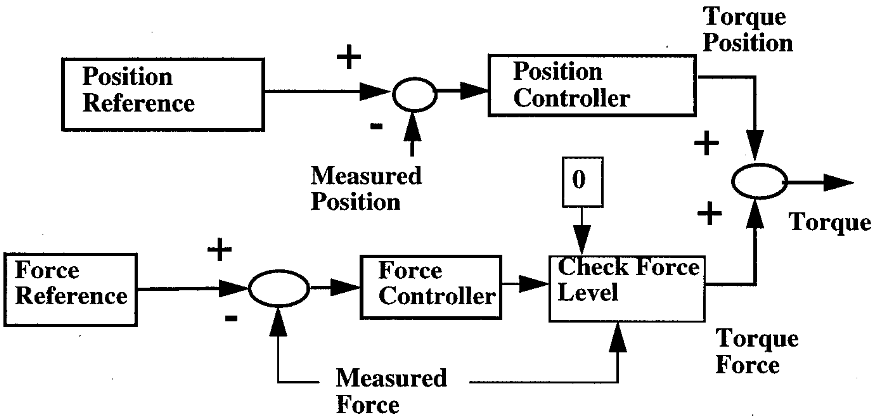


Figure 14.1- Baseline Hybrid Force and Position Control Scheme

For this research, the thresholds are experimentally tuned to 7 gf for initial contact and 4 gf after that. The reason 0 gf isn't chosen is due to the fact that the force sensor is an electrical device, and has an electrical zero which varies slightly throughout the run. In addition, the movement of the arm generates a small amount of dynamic force. Once contact is made, a lower threshold is allowed to allow a greater chance of maintaining contact, especially after the initial impact.

The impact can be described by basic dynamics equations to give some a priori insight. Assuming perfectly elastic collisions, the following equation [50] describes the impact of two rigid bodies.

$$m_1 u_1 + m_2 u_2 = m_1 v_1 + m_2 v_2 \tag{14.1}$$

where

u_1, u_2 = speeds of bodies 1 and 2, respectively, before impact

v_1, v_2 = speeds of bodies 1 and 2, respectively, after impact

m_1, m_2 = massed of bodies 1 and 2, respectively

In this case of one of the bodies being fixed, like in an impact control application, the velocity of the fixed object is always zero. Thus, the equation simplifies to

$$m_1 u_1 = m_1 v_1 \tag{14.2}$$

In words, the impact is dependent upon both the speed and the mass of the manipulator. The resulting velocity, of course, is in the opposite direction. A higher momentum of the manipulator as it impacts with the surface results in a greater tendency to lost contact. Thus, the dynamics say that larger masses and/or higher speeds will result in a more difficult touchdown.

14.1 Baseline Touchdown Results

Analyzing the baseline system gives insight into possible touchdown solutions. All the tests in this chapter are run at a 1 ms sampling time. The manipulator is run at various speeds into a rigid metal surface, without knowledge of the exact surface location. This is accomplished by commanding a position which would take the manipulator somewhere into the surface. It is assumed that the general location of the surface is known.

The speed tests show that impact force is directly proportional to impact speed, as expected. First, no force control is applied to observe the resulting impact force with just position control. Then, the same speeds are repeated and force control is applied upon impact (defined as the instant 7 gf is first achieved). Figure 14.2 shows that when force control is applied, the impact force is slightly higher than with pure position control. This is likely due to the force control helping push the arm further into the surface initially while trying to achieve the commanded force level.

Next, with the hybrid control applied, the speed at which successful touchdown (maintaining surface contact) occurs is examined. For all of these speed tests, a 50gf reference level is used (since this is the level used in previous research on the arm). At a speed of 0.6 cm/s, the touchdown is always smooth with no discernible bounce. The plot of this experiment is shown in Figure 14.3. Notice that the touchdown recoil is low after an initial contact, and that the impact force is about 20 gf. Experiments showed that the higher the impact force, the greater the recoil, as expected.

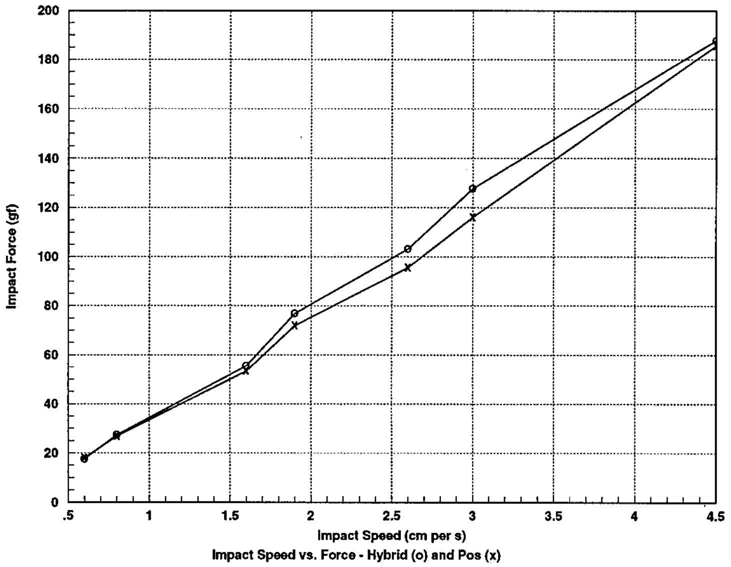


Figure 14.2- Impact Force vs. Impact Speed

At 0.8 cm/s, the force level is maintained about 60% of the time, and there is a noticeable impact recoil. When the speed is increased to 1.6 cm/s, contact is never maintained and the naked eye can discern a loss of contact with the surface for 1 bounce. As the impact speed is increased further, several bounces are required before contact is maintained. The plot for the 1.6 cm/s experiment is shown in Figure 14.4. This plot gives insight into the baseline limit for maintaining contact. The units on all the experimental plots are volts, Newton-meters (for the torques), and gf for the forces. Examination of this figure shows that the initial impact builds up to just past 50 gf in the first 0.1 second, and rapidly drops off for about the next 0.2 seconds causing the arm to lose contact with the surface. This is due to the impact with the surface

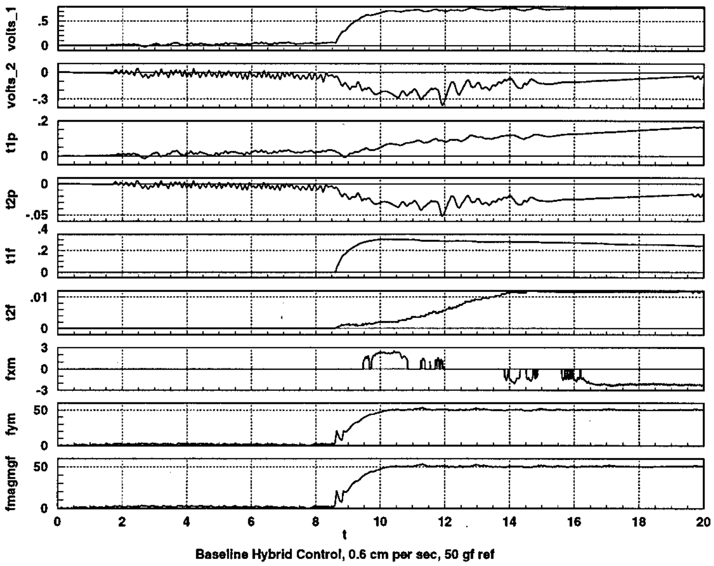


Figure 14.3- 0.6 cm/s Touchdown with 50 gf Reference, Baseline Performance

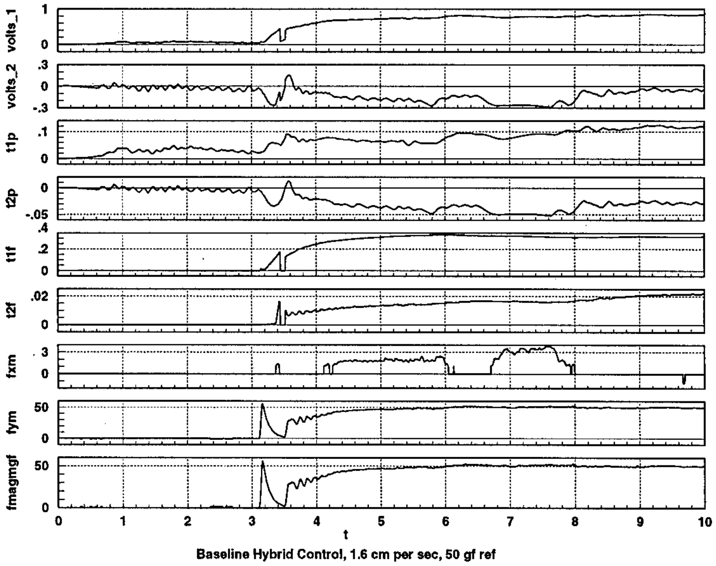


Figure 14.4- 1.6 cm/s Touchdown with 50 gf Reference, Baseline Performance

The system bandwidth is not high enough to respond to this impact, even though once in contact these controllers give excellent results. Thus, it appears that some kind of "bandwidth boost" is needed for about the first 0.3 - 0.5 seconds after impact.

Next, several other factors which affect the touchdown performance are examined. Specifically, the commanded force level and position reference are analyzed. Force reference commands of 20 gf, 50 gf, and 150 gf are used at a 0.8 cm/s speed. For the 20 gf level, there was degraded performance and contact was never maintained. At the 150 gf level, contact was maintained on every attempt. The 50 gf was used in the original baseline test discussed earlier. Thus, increasing the reference force command enabled a better touchdown performance.

When the position reference was changed from somewhere in the surface to the exact location measured at touchdown, there was a noticeable performance degradation. Examine the baseline case shown in Figure 14.5, where at 0.8 cm/s contact is maintained sometimes. Even when the contact is lost, it is only for a short time.

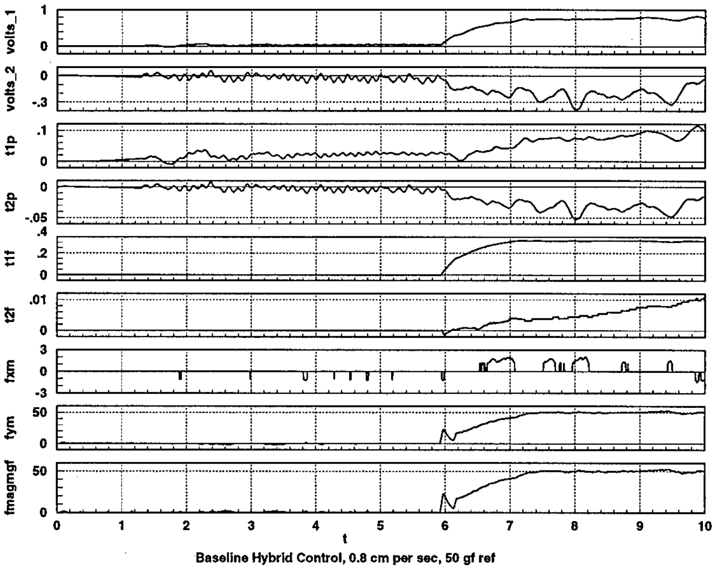
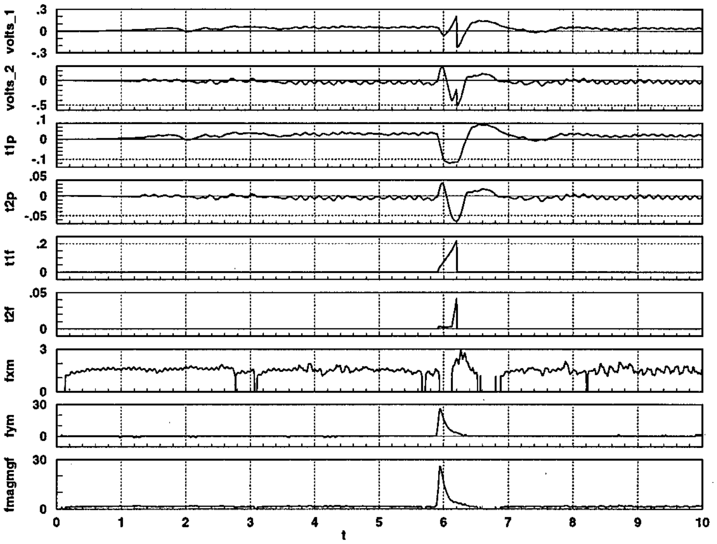


Figure 14.5- 0.8 cm/s Touchdown with 50 gf Reference, Baseline Performance

Contrast this with Figure 14.6, where the position reference is changed from into the surface to the exact touchdown location upon impact. Notice that the recoil is amplified because now the position torques oppose the force torques. Thus, providing a reference slightly into the surface helps the surface touchdown. This is due to the fact that with a position reference into the surface, the position torques are in the same direction as the force torques. The overall baseline test observations are summarized in Table 14.1.



Baseline Hybrid Control, 0.8 cm per sec, catch xy-td,50 gf ref

Figure 14.6- 0.8 cm/s Touchdown with 50 gf Reference, Surface Reference

Table 14.1- Factors Affecting Touchdown

Factor	Effect
Increased Speed	Decreased Performance
Increased Force Reference	Increased Performance
Position Reference Into Surface	Increased Performance

14.2 Touchdown Control Strategies

By analyzing the baseline results, improvement areas stand out for both the position and force schemes. For the position reference, provide a location into the surface so that the position torques are in line with the force torques. Maintain this offset for at least 1 second after impact until firm contact has been established. Experience has shown that a moderate offset of about 1 inch is adequate for this application.

The force aspect of the hybrid scheme can be improved several ways. First, providing a temporary "bandwidth boost" immediately after impact when it is most needed seems logical. Also, providing a temporary "force boost" to the force reference command holds promise. The modified hybrid control structure to include these modifications are shown in Figure 14.7.

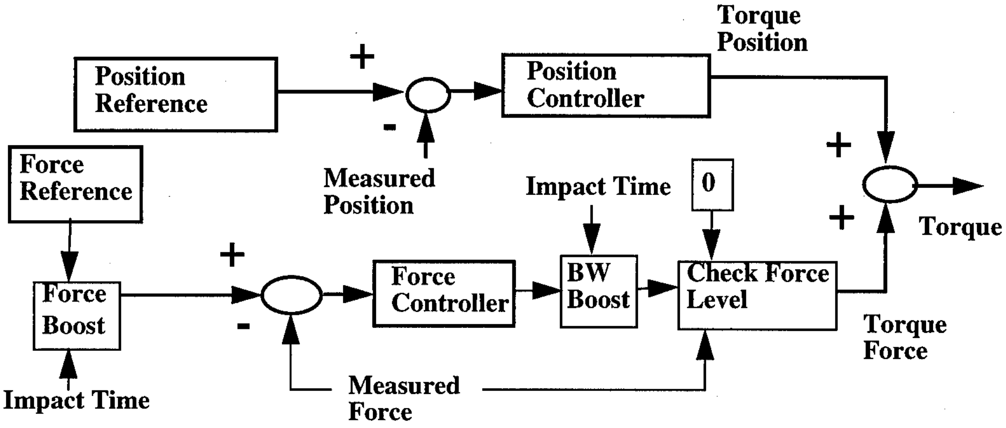


Figure 14.7- Modified Hybrid Force and Position Control Scheme

A simple design procedure is discussed to improve surface touchdown tasks. There are four degrees of freedom for the surface touchdown task:

1. Vary length of time that the force boost is applied after touchdown.
2. Vary multiplication factor for the force boost.
3. Vary length of time that the bandwidth boost is applied after touchdown.
4. Vary multiplication factor for the bandwidth boost.

These four factors are varied iteratively until an acceptable touchdown response is achieved. The gain factors and length of application for both parts of the scheme don't have to be the same, although they may be identical if the user desires. Notice that the force and position controllers don't have to be redesigned, which is an advantage to this approach.

14.3 Experimental Implementation

Using this scheme, a force boost of 2 is applied for 0.3 seconds after impact and a bandwidth boost of 2 is applied for 0.5 seconds. This is applied to the nominal test case of 1.6 cm/s impact speed and a 50 gf commanded force level, and the results are shown in Figure 14.8. The same force and position controllers are used, and only the touchdown modifications are different between this test and the baseline. Notice that the force level never drops below 20 gf, and that contact is maintained. Compare this to the nominal case of Figure 14.4, where contact is never maintained at this speed and commanded force level. Thus, this method provides a noticeable improvement over the nominal case.

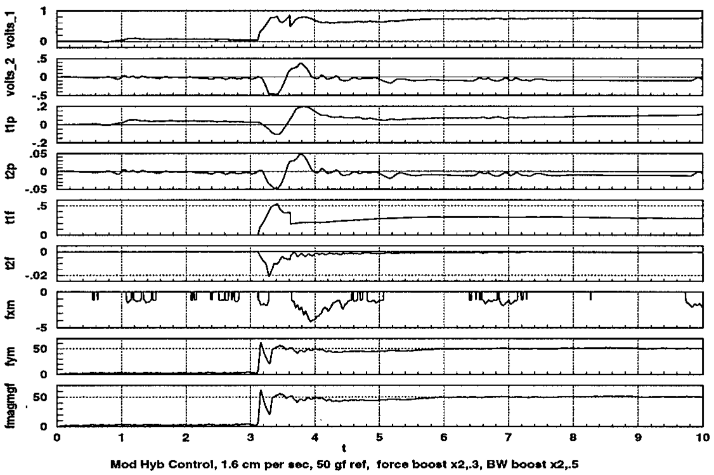


Figure 14.8- Experimental Results, Modified Touchdown Scheme

15 The Complete Hybrid Task

In a typical manufacturing task, there are several stages. The tool first starts in some arbitrary location and then it makes contact with a work surface. Once in contact, hybrid control is applied to control both position and force of the tool. Finally, the tool must be removed from the surface and returned to the initial arbitrary location. This is known as the complete hybrid task in this work.

An analytical approach to the complete hybrid task divides it into 6 distinct phases. Figure 15.1 shows the phases and where they occur during the complete hybrid task.

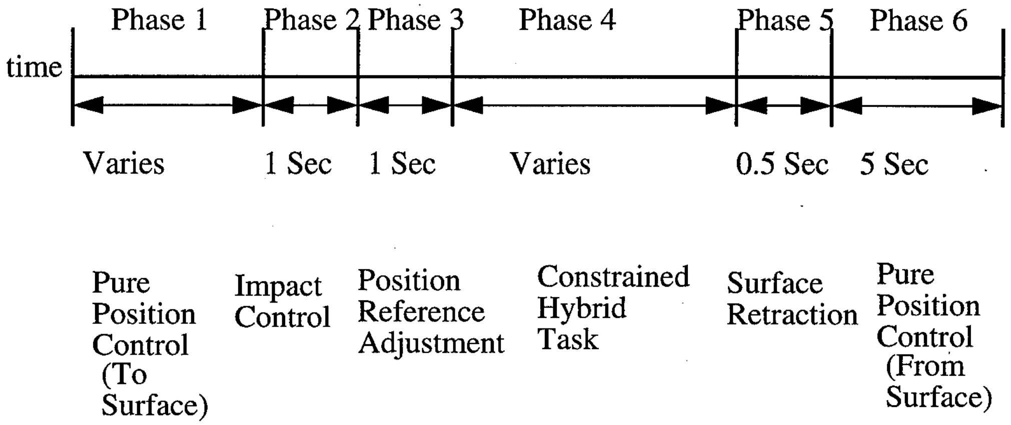


Figure 15.1- Complete Hybrid Task Description

During Phase 1, the tool is moved from the initial arbitrary location towards the surface. The position reference must be in the general direction of the surface, and must be somewhere into the surface. Otherwise, the tool would never make contact with the surface. During this phase, only the position control is applied.

Phase 2 starts when the tool makes contact with the surface, defined as the first instant that 7 gf is measured by the force sensor. This value is chosen since the force sensor is an electrical device with an electrical zero which may vary slightly. Also, when the arm is in motion, there are some dynamic forces present. 7 gf is high enough to distinguish between dynamic forces and electrical zeros versus an actual impact for this application. The specifics of the surface touchdown or impact control are described in a previous chapter. This phase lasts for 1 second, which allows enough time to maintain good contact between the surface and the tool.

After Phase 2 ends, Phase 3 allows transition from the current reference position to the position where the manipulator first made contact with the surface. This allows the complete hybrid task to work when the exact position of the surface is not known. When the impact is initially detected (at 7 gf), the end-point position is saved. Then, at the start of Phase 3, the position reference is changed to allow a smoother transition for surface tasks.

Some experimental observations for Phase 3 are presented now to give some physical insight into why this phase is required. If the initial reference is very far from the surface, a large step command is essentially generated which the manipulator can't respond to instantaneously. Thus, to ease the transition, the reference at the surface is applied and the manipulator is given 1 second to stabilize before starting the constrained hybrid control task. There is a tradeoff between how far the initial position reference command is given into the surface and the Phase 3 performance. The further inward the initial reference is given, the better the impact stage response since the position control aids the force control. However, a greater difference between the reference and the wall results in a more difficult transition to the constrained hybrid control stage due to a larger step change in the position reference.

Additionally, link 2 has the ability to move and follow its reference, while link 1 is physically constrained by the wall (through link 2 and the roller mechanism). Thus, upon impact with a reference far into the surface, the tool has a tendency to roll to the right. The further the reference is into the surface, the further to the right the tool moves. Of course, since Phase 2 lasts only 1 second, the full effect is not seen. For this application, 1 second for Phase 3 works well, but the length of time could be varied for another system.

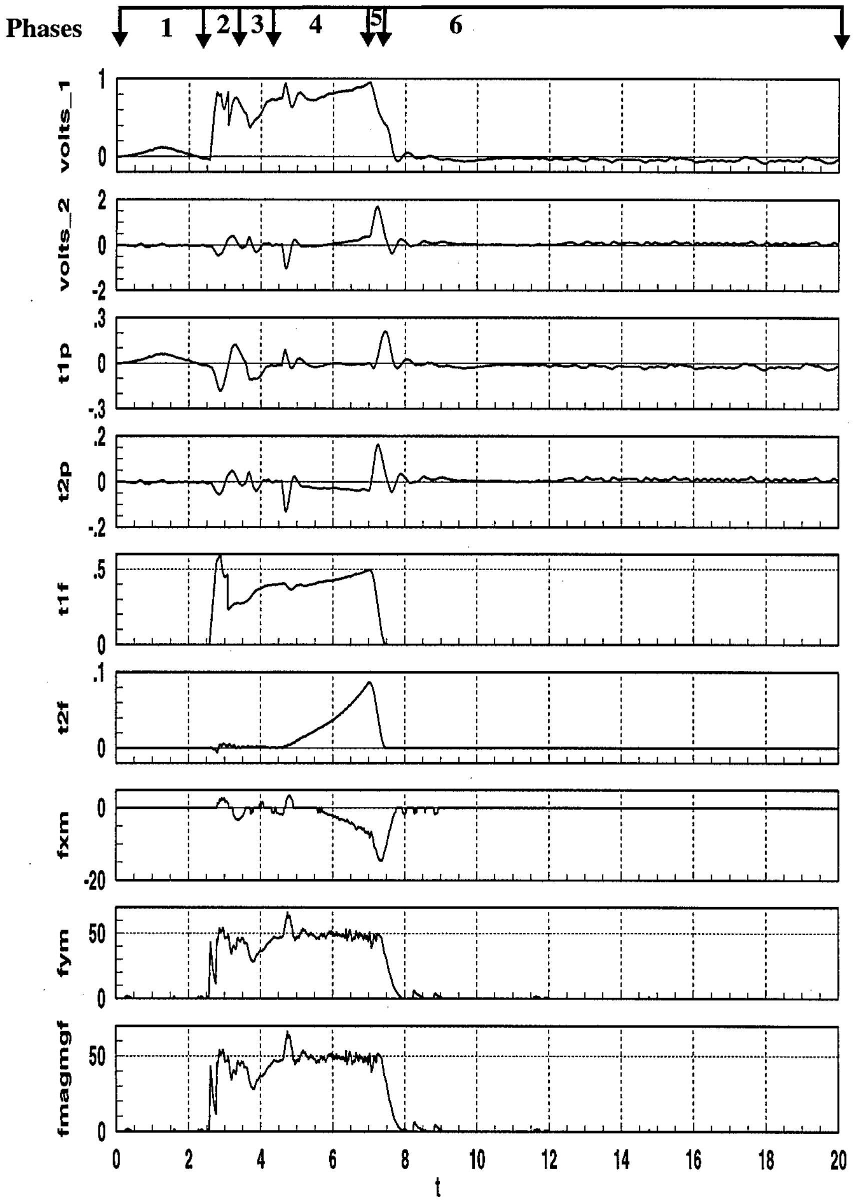
Once the tool is in contact with the surface and is at a known reference location, Phase 4 begins. This is the hybrid position and force scheme described in earlier chapters. The surface position may be either known or unknown, depending on the application. The unknown surface following algorithm described in previous chapters allows an unknown constrained surface to be followed. Of course, in many manufacturing applications, the exact surface shape is known, and can be used during this phase. This is the most important phase of the complete hybrid task in the sense that the actual work is accomplished during this phase.

When the constrained hybrid task is completed, Phase 5 of the complete hybrid task can begin. After a specified time, the tool must be withdrawn from the surface. Experimental analysis showed that an instantaneous cutout of the force control results in an undesirable transient response of the manipulator. This transient manifested itself in a discernible vibration which could be seen with the naked eye. Thus, Phase 5 uses the same spline function used for the position reference to spline both torques associated with the force controllers from their values at the start of Phase 5 down to zero. The time of this spline is a design variable, and this application used a value of 0.5 seconds.

Finally, after the force torques are splined down to zero, Phase 6 begins. The current end-point position is saved, and the initial starting position is known. This allows a position reference spline from the current location to the initial location. This phase is purely position control, just like the Phase 1. The time taken for this phase depends on the location of the end-point relative the starting location. If the two points are far apart, more time may be required for this last phase. For this application, 5 seconds worked well for all of the experiments performed.

This complete task should work for different controllers and for arbitrary shapes. The two controllers used for the testing are the full 46th-order LQG hybrid position and force controller and the 10th order robust reduced-order SANDY controller. All of the strip charts have units of volts (for voltages), Nm (for torques), and gf (for forces).

The LQG controller is applied to a straight line without knowledge of the line's shape or exact location, and the results are shown in Figure 15.2. The commanded speed for the surface algorithm is 1 cm/s and the commanded force is 50 gf for the constrained motion hybrid task. Phase 1 lasts approximately 2.5 seconds, until contact with the wall occurs. Notice that there is an initial recoil in the force level due to the collision with the wall at about 2.5 seconds, corresponding to the start of Phase 2.



Retract, LQG, .005 offset, 1 s hold, 1 s wall, unk line, 1 cm per sec

Figure 15.2- LQG, Unknown Line Strip Chart

Phase 3 begins at about 3.5 second, causing a dip in the force level due to the transition of the position reference to the wall position. At about 4.5 seconds, the unknown surface following algorithm starts, and there is a small increase in the measured force corresponding to the start of Phase 4. Notice that during Phase 4, the force level is maintained at the commanded 50 gf level while tracking the unknown surface. After 7 seconds, Phase 5 begins and notice that the force torques are gradually splined down to 0 Nm after an additional 0.5 second. Finally, Phase 6 involves the position slew back to the initial position. Notice that the force level drops to zero after the tool leaves the surface.

The corresponding end-point location is shown in Figure 15.3. The tool starts about 0.025 meters from the surface (about 1”), and the reference is a line in the general direction of the surface. The dashed line is the reference signal and the solid line corresponds to the end-point location. The x’s correspond to 0.5 second intervals, and allow the end-point position to be related back to the strip chart.

Notice that the reference signal changes from somewhere into the surface to the point of initial contact. When the unknown surface algorithm is engaged, observe that a relatively straight line reference is given. After Phase 5 is completed, notice that reference changes to the current location and the reference is a straight line back to the initial starting point

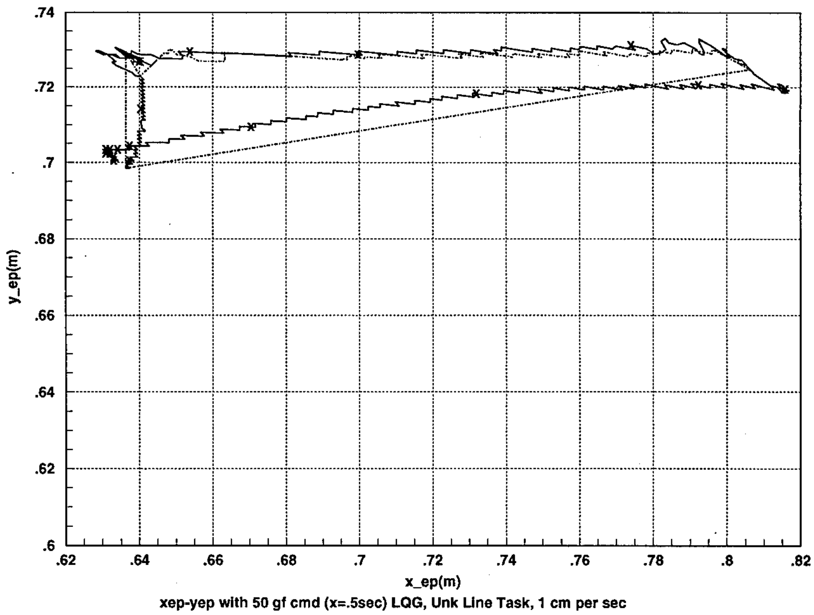
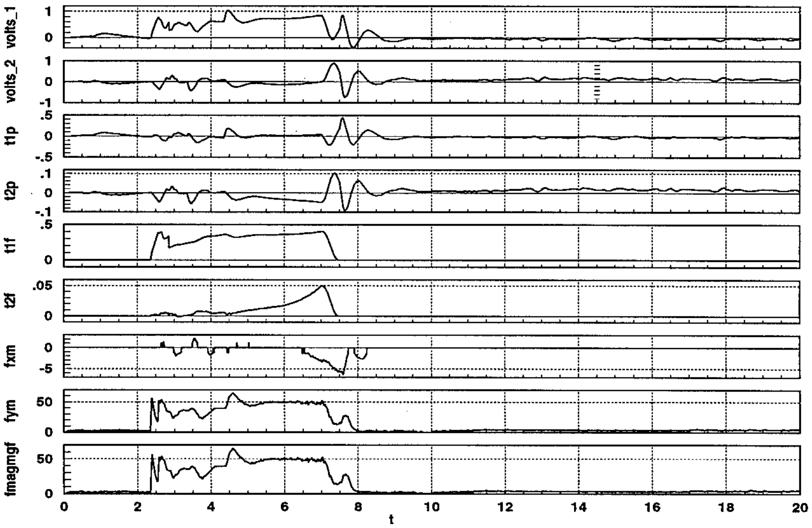


Figure 15.3- LQG, Unknown Line End-Point Position

The next experiment tests the reduced-order SANDY controller at a rate of 0.7 cm/s and 50 gf commanded for the same straight line surface. The corresponding strip chart is shown in Figure 15.4. The same trends are shown for the SANDY controller as for the LQG controller, namely that transition between the phases can be seen. The SANDY controller also provides acceptable performance by maintaining contact after initial impact and then successfully applying the commanded 50 gf while tracking the unknown surface.

The corresponding SANDY end-point plot is shown in Figure 15.5. The same general trends for the reference are visible here as in the LQG test. There is a little more movement to the left and right on the surface after initial contact for the SANDY controller, but after that the line is followed well. The final phase of position control involves the same line reference from the end-point location at the start of Phase 6 to the initial starting location.

The next check of the complete task is to follow an unknown arc surface at an unknown location. First, the LQG controller is applied at the same 1 cm/s tracking speed and 50 gf commanded force level for 10 seconds. Figure 15.6 shows that contact is made at about 2.5 seconds, and the same transitions between the phases are vis-



Retract, SAN, .005 offset, 1 s hold, 1 s wall, unk line, .7 cm per sec

Figure 15.4- SANDY, Unknown Line, Strip Chart

ible. Once the constrained motion phase starts, the force is maintained at the commanded level while the surface is followed. Finally, the tool is returned to the initial position through pure position control.

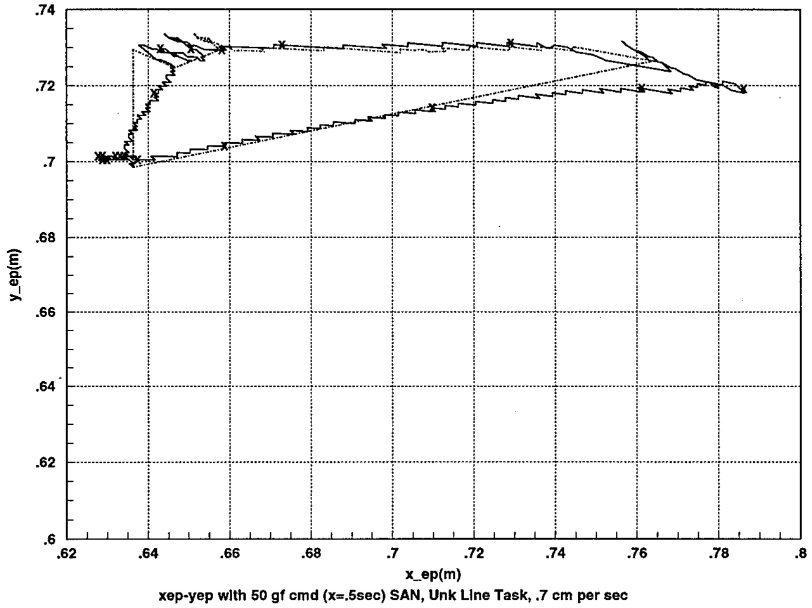


Figure 15.5- SANDY, Unknown Line, End-Point Position

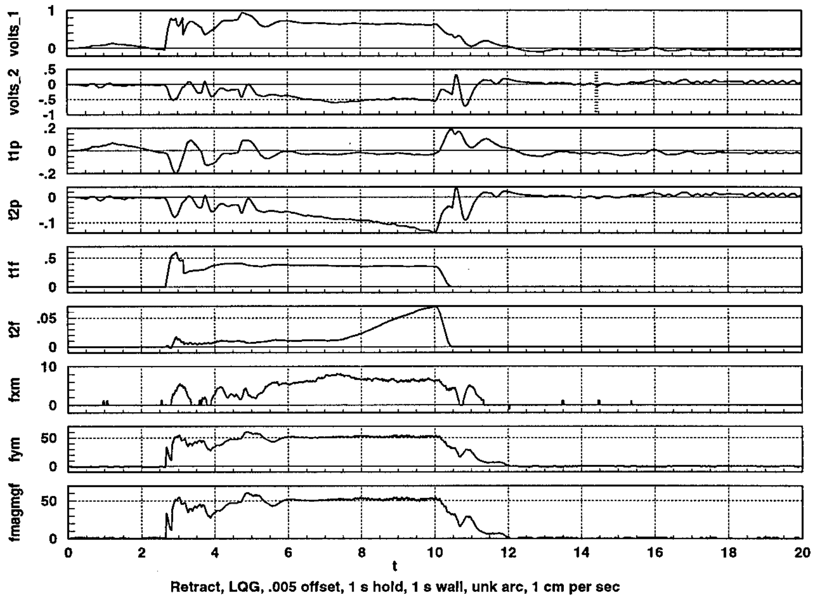


Figure 15.6- LQG, Unknown Arc, Strip Chart

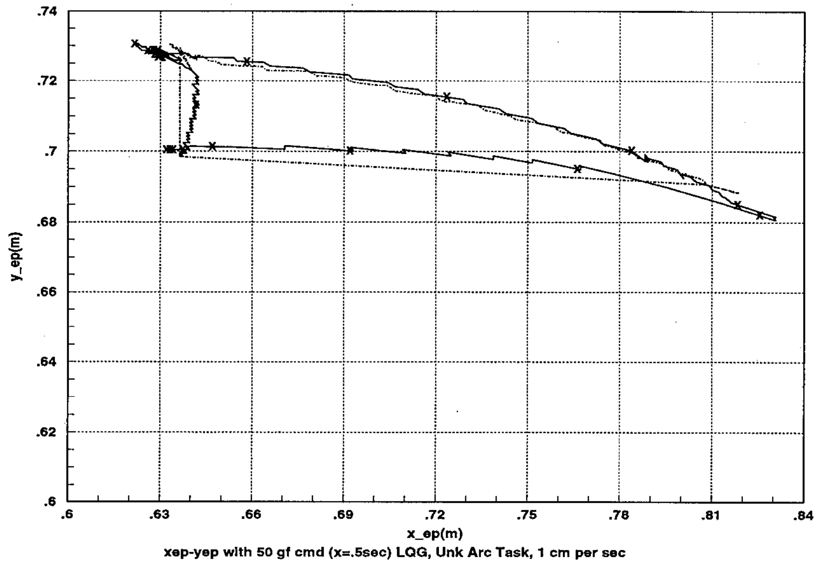


Figure 15.7- LQG, Unknown Arc, End-Point Position

The corresponding end-point plot is shown in Figure 15.7. Once again, the reference is into the surface and switched to the location of initial contact. The surface following algorithm then correctly provides a reference for following an arc.

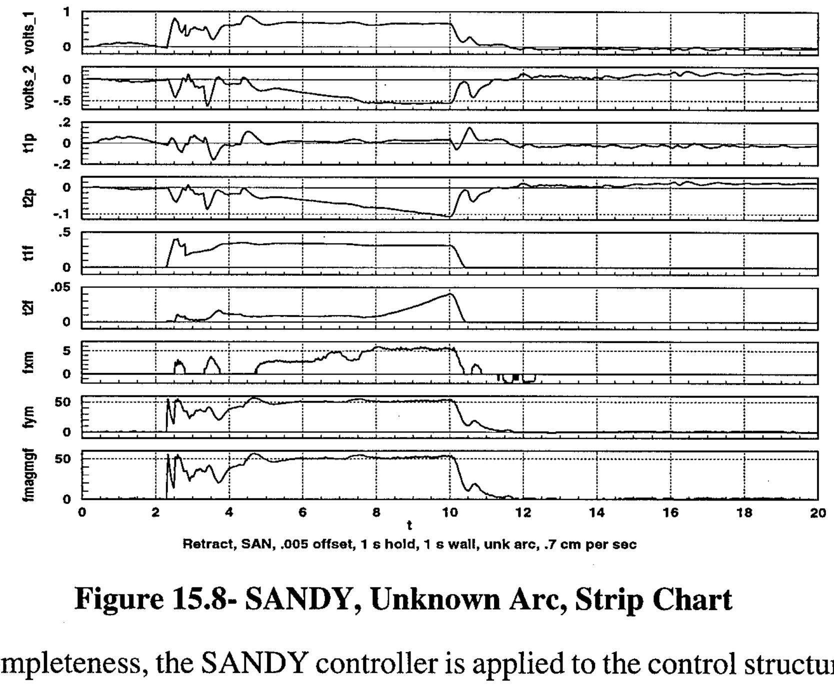


Figure 15.8- SANDY, Unknown Arc, Strip Chart

For completeness, the SANDY controller is applied to the control structure for an unknown arc at an unknown location. The experiment is run for a 50 gf command and a rate of 0.7 cm/s for the surface following algorithm for 10 seconds, as shown in Figure 15.8. The SANDY position controller moves the arm a little more quickly, and thus a higher initial recoil occurs. After transitions between the phases, the SANDY controller also provides good performance for the constrained motion phase. Finally, the end-point contact with the surface is ended and returned to the starting position.

The corresponding end-point plot is shown in Figure 15.9. Once again, the same trends noticed for the LQG controller are present. The end-point moves and toward the surface, makes contact with the surface, follows the unknown arc, and then retracts to the initial starting position.

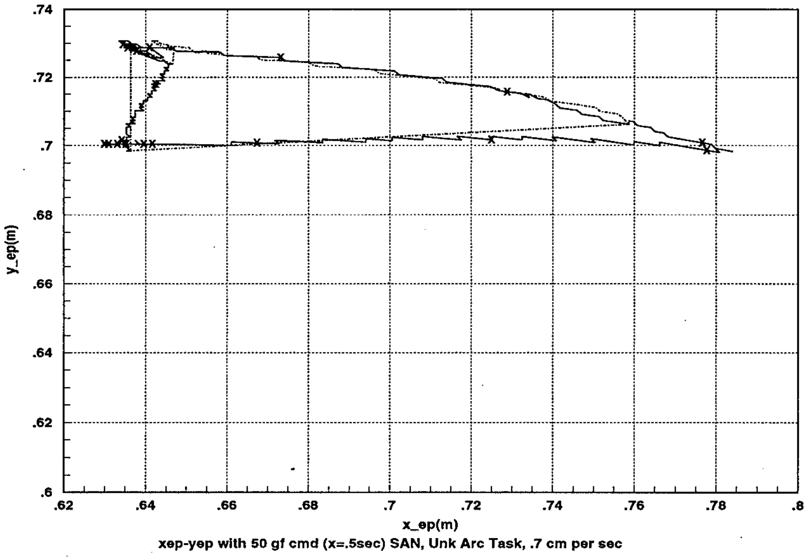


Figure 15.9- SANDY, Unknown Arc, End-Point Position

Thus, the complete hybrid task structure is experimentally verified. The structure works for various unknown surfaces as well as full and reduced-order controllers. The constraint is that the impact speed is limited by the physical bandwidth of the actuators and the system. However, the approach speed in the initial phase can always be adjusted to provide an acceptable touchdown speed. The big advantage of this approach is that the exact location and geometry of a surface is not required a priori. The only requirement is that the general direction and general location of the surface be known. This is because the tool does need to physically make contact with the surface.

16 Conclusion

16.1 Summary

This research represents a consolidation of numerous tasks ranging from developing the force and position models of a two-link flexible manipulator to the design and test of robust reduced-order hybrid position and force controllers. In all 12 position controllers and 12 force controllers were designed and experimentally evaluated for a total of 24 different controller designs. This does not include the combinations tested for hybrid superposition control.

The initial goal of the research was to prove that robust reduced-order controllers could provide comparable performance with full-order controllers. The experimental results for position, force, hybrid superposition, and hybrid complete optimization showed that indeed this is the case.

Another goal of the research was to develop a hybrid framework compatible with a typical manufacturing task. Namely, start at an arbitrary initial condition, move to a manufacturing surface, make contact, and then perform a constrained motion hybrid task while maintaining a constant force. Finally, retract from the surface gracefully and return to the initial position. The complete hybrid framework developed in this dissertation allows the aforementioned task to be accomplished using the *same controllers* throughout the task. In addition, the framework allows the situation where the manufacturing workpiece shape and exact location is not known.

Connecting both major thrusts of the dissertation together involved comparing the performance of the full-order LQG controllers with the robust-reduced order SANDY controllers in the complete hybrid control scheme. Once again, the reduced-order controllers provided comparable performance to the full-order controllers.

16.2 Design Lessons Learned

Since this was an application-oriented dissertation, insight into the design and implementation of hybrid controllers was obtained. Some lessons learned are discussed in the following section.

16.2.1 Hardware Variability

Since the two-link arm (TLA) is a real physical device, it was noticed that the performance varied some from day to day. There are several reasons for this. First, the TLA floats on a flat table through air pads to provide a "low friction" environment. However, depending on how the arm was balanced, the actual friction level varied. This was especially noticeable after the links were changed out to accommodate other experiments.

In addition, the motors are DC torquers which possess effectively zero backlash. However, since an integral component of these motors is a permanent magnet, there was always some residual magnetism. When the arm was moved, it had a tendency to move back toward the initial location when the control torques were removed. Also, this effect was amplified when low torques were applied.

The lesson learned here is that when comparing different controllers, it is very important to compare them at about the same time. This assures that they are acting on the same plant. Of course, since any physical system is actually nonlinear, there will always be some variability in performance. If the plant variation is drastic from day to day, a robust controller can be designed to work for all of the plant conditions.

16.2.2 Optimization Observations

Several very important design observations were made regarding controller optimization for flexible structures. First, when optimizing controllers for a flexible structure, it is very important to have at least one plant condition to excite all frequencies. Namely, have a plant with a white noise input directly into the control input, similar to the LQG/LTR fictitious noise input. In early optimization attempts when only a filtered command and a pure step input was used, the optimization tended to ignore the high frequency modes. Thus, the rigid body modes were optimized and gave good high bandwidth performance, but they excited some of the flexible modes. Placing the white noise (impulse input) as one plant case allowed the optimization to consider the flexible modes as well as the rigid body modes.

Another key element was a variable damping constraint. With only hub actuators and sensors available, there was no way to control the flexible modes. Thus, a universal damping constraint could not be used. Therefore, the optimization had no way to constrain the damping of the rigid body modes. By adding a damping constraint below the first flexible mode frequency but above the frequency of the rigid body modes, the optimization could effectively add damping to the rigid body modes without trying to control the flexible modes.

Another observation is that the optimization worked better with a good initial guess, as logic would dictate. Thus, it is important to use all the information available for the initial guess. Sources for initial guesses include simple PID or classical controllers.

Finally, there was a trade-off between the number of plants chosen for a robust design and the computation time. The designer should consider adding only enough plants to cover the design objective. Plants that are between two other plant cases should be left out. For example, if the designer has models for a light, medium, and heavy tip, only the light and heavy models need to be used.

16.2.3 Logical Design Steps

Many times, one of the most vexing questions facing a designer is "Where do I start?" The experience gained in this dissertation provides useful guidelines for hybrid position and force control designs.

1. Design a single plant position controller.
2. Design a robust position controller starting from the original position controller (if needed).
3. Design a single plant force controller.
4. Design a robust force controller starting from the original force controller (if needed).
5. Design a hybrid position and force controller starting from an existing position and force controller design.
6. Design a robust reduced-order hybrid position and force controller starting from the original hybrid position and force controller design.

16.3 Recommendations for Future Research

One of the interesting things about conducting dissertation research is that normally for every question answered, several more arise. Given a finite amount of time, there are only so many things which can be accomplished in one dissertation. Bearing that in mind, here are some recommendations for future research.

First, end-point position sensing should be added to the design effort. This would allow both collocated and non-collocated control of position. Keeping in mind that with a flexible manipulator, the end-point position can only be approximated through the forward kinematics makes this a logical extension. In addition, attempting hybrid control on more flexible links would provide a more challenging design problem.

Sensor blending between the end-point sensor and the hub angle sensors would also provide interesting research. Also, multiple sensors open the door for fault detection and isolation research on a two-link flexible manipulator.

Finally, an area that provides some interesting questions is in terms of the interaction between the force and position controllers. There is a trade-off between the force and position controller interaction in terms of bandwidth. Bandwidth separation was used to provide acceptable interaction between the force and position controllers. The force control was given higher bandwidth to maintain a desired force level while in contact with the surface. However, it was also discovered that for sharp inverted arcs that the force controller would apply torques which would tend to move the end-point and the position controller did not have the bandwidth to hold the end-point at the desired location. Somehow, if the relation between the two bandwidths could be quantified for different shapes, it would aid in future designs.

Bibliography

1 An, C., and Hollerbach, J., "Dynamic Stability Issues in Force Control of Manipulators", *Proc of the IEEE Int Conf on Robotics and Automation*, pp. 890-896, 1987.

2 Anderson, B.D.O. and Moore, J.B., "Linear Optimal Control", Prentice Hall Inc., Inglewood Cliffs, New Jersey, 1971.

3 Ballhaus, W., "Experiments in High-Performance Control of a Multi-Link Flexible Manipulator With a Mini-Manipulator", Ph.D. Dissertation, Department of Aeronautics and Astronautics, Stanford University, Apr 1994.

4 Bodur, M. and Sezer, M., "Adaptive Control of Flexible Multilink Manipulators", *Int J. of Control*, Vol. 58, No. 3, pp.519-536, 1993.

5 Bossert, D., Evers, S., Ly, U., and Vagners, J., "Establishment of a Flexible Two-Link Manipulator System at the University of Washington and Experimental Validation with PID and SANDY Controllers", University of Washington Internal Report, 1994.

6 Bossert, D., Lamont G., Leahy, M., and Horowitz, I., "Model-Based Control with Quantitative Feedback Theory", *Proc of the IEEE Int Conf on Robotics and Automation*, May 1990.

7 Bossert, D., Ly, U., and Vagners, J., "Evaluation of Reduced-Order Controllers on a Two-Link Flexible Manipulator", *Proceedings of the 1995 American Control Conference*, pp. 3339-3345, Jun 1995.

8 Bossert, D., Ly, U., and Vagners, J., "Robust Reduced-Order Control of a Flexible Two-Link Manipulator," *Proceedings of the 1995 IASTED International Conf on Robotics and Manufacturing*, pp. 328-331, Jun 1995.

9 Bossert, D., Ly, U., and Vagners, J., "Experimental Evaluation of Robust Reduced-Order Hybrid Position/Force Control on a Two-Link Flexible Manipulator", *Proceedings of the 1996 IEEE Robotics and Automation Conference.*

10 Bossert, D., Ly, U., and Vagners, J., "Experimental Evaluation of a Hybrid Position and Force Surface Following Algorithm for Unknown Surfaces", *Proceedings of the 1996 IEEE Robotics and Automation Conference.*

11 Bossert, D., Ly, U., and Vagners, J., "Experimental Robust Reduced-Order Hybrid Position/Force Control for Two-Link Flexible Manipulators", submitted to the *IEEE Transactions on Control Systems Technology.*

12 Cai, L., and Song, G., "Robust Position/Force Control of Robot Manipulators during Contact Tasks", *Proceedings of the 1994 American Control Conference*, pp. 216-220, Jun 1994.

13 Carusone, J. and D'Eleuterio, G., "Tracking Control for End-Effector Position and Orientation of Structurally Flexible Manipulators", *Journal of Robotic Systems*, Vol 10, No. 6, pp. 847-870, 1993.

14 Chen, Y. and Yeung, K., "Sliding-Mode Control of Multi-Link Flexible Manipulators", *Int J. of Control*, Vol. 54, No. 2, pp. 257-278, 1991.

15 Craig, J., "Introduction to Robotics Mechanics and Control", Addison-Wesley Publishing, New York, 1989.

16 Dawson, D., Lewis, F., and Dorsey, J., "Robust Force Control of a Robot Manipulator", *Int Journal of Robotics Research*, Vol. 11, No. 4, pp. 312-319, Aug 1992.

17 D'Azzo, J. and Houpis, C., "Linear Control System Analysis and Design", Mcguire-Hill Book Company, New York, 1988.

18 Demey, S., and De-Schutter, J., "Enhancing Surface Following With Invari-

ant Differential Part Models", *Proc of the 1994 IEEE Int Conf on Robotics and Automation*, pp. 668-673, Vol. 1, May 1994.

19 Denoyer, K. and Kwak, M., "Dynamic Modeling and Vibration Suppression of a Slewing Active Structure Utilizing Piezoelectric Sensors and Actuators", *Smart Structures and Intelligent Systems*, pp.882-894, 1993.

20 Doyle, J. "Guaranteed Margins for LQG Regulators," *IEEE Transactions on Automatic Control*, Vol. AC-23, No. 4, pp. 756-757, Aug 1978.

21 Eisler, G., Segalman, D., Robinett, R., "Approximate Minimum-Time Trajectories for Two-Link Flexible Manipulators", *Proceedings of the American Control Conference*, pp. 870-875, May 1990.

22 Evers, S, "The UWCSL Two-Link Manipulator Facility: System Description", University of Washington Internal Report, Jan 1994.

23 Feliu, V., Rattan, K., and Brown, H., "Control of FLexible Arms with Friction in the Joints", *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 4, pp. 467-475, Aug 1993.

24 Feng, G. and Palaniswami, M., "An Adaptive Control Algorithm With Application to Single-Link Flexible Manipulators", *Int J. of Robotics and Automation*, Vol. 7, No.3, pp. 111-116.

25 Fisher, W., and Mutjaba, M., "Hybrid Position/Force Control: A Correct Formulation", *Int. J of Robotics Research*, Vol. 11, No. 4, pp. 299-311, Aug 1992.

26 Gangsaas, D., Ly, U., and Norman, D., "Practical Gust Load Alleviation and Flutter Suppression Based on an LQG Methodology", *Proceedings of the AIAA 19th Aerospace Sciences Meeting,* Jun 1981.

27 Hein, C., and Meystel, A., "A Genetic Technique for Robotic Trajectory Planning", *Telematics and Informatics*, Vol. 11, no. 4, pp. 351-54, Fall 1994.

28 Houpis, C. and Lamont, G., "Digital Control Theory, Hardware, Software", New York: McGraw-Hill Book Company, 1985.

29 Huang, H., and Lin, M., "Robust Force Control for Robotic Manipulators", *Int Journal of Control*, Vol. 56, No. 3, pp. 631-653, 1992.

30 Kane, T.R. and D.A. Levinson, "Formulation of Equations of Motion for Complex Spacecraft", *AIAA Journal of Guidance, Control, and Dynamics*, 3(2):99-112, March-April 1980.

31 Khatib, O. and Burdick, J., "Motion and Force Control of Robot Manipulators", *Proc of the IEEE Int. Conf on Robotics and Automation*, pp. 435-441, 1986.

32 Koivo, A. and Lee, K., "Self-Tuning Control of a Two-Link Manipulator With a Flexible Forearm", *Int J. of Robotics Research*, Vol. 11, No. 4, pp. 383-395, Aug 1992.

33 Khorrami, F., Sandeep, J., Tzes, A., "Experiments on Rigid Body-Based Controllers with Input Preshaping for a Two-Link Flexible Manipulator", *IEEE Transactions on Robotics and Automation,* Vol. 10, No. 1, pp. 55-65, 1994.

34 Kraft, R., "Experiments in End-Point Control of a Flexible Robot with a Mini-Manipulator", Ph.D. Dissertation, Department of Aeronautics and Astronautics, Stanford University, May 1989.

35 Kwakernaak, H. and Sivan, R., "Linear Optimal Control Systems", Wiley-Interscience, 1972

36 Kwan, C., Yesildirek, A., and Lewis, F., "Robust Force/Motion Control of Constrained Robots Using Neural Networks", *Proceedings of the 33rd Conference on Decision and Control*, pp. 1862-1867, Dec 1994.

37 Hogan, N., "Impedance Control of a Robotic Manipulator", *Proc of the ASME Winter Annual Meeting*, 1981.

38 Hyde, J., and Cutkosky, M., "Contact Transition Control: An Experimental Study", *Proc of the IEEE Int. Conf on Robotics and Automation*, Vol. 1, pp. 363-368, 1993.

39 Laub, A.J., Computation of Balancing Transformations, Proc of JACC Vol. 1, paper FA8-E, 1980.

40 Leahy, M., Bossert, D., and Whalen, P., "Robust Model-Based Control: An Experimental Case Study", *Proc of the IEEE Int Conf on Robotics and Automation*, May 1990.

41 Lin, Z., Saberi, A., and Chen, B., "Linear Systems Toolbox", Washington State University Technical Report EE/CS 0097, Washington State University, Pullman, WA.

42 Liu, D., Liu Z., Yin, C., and Wu C., "Robot Learning Control Based On Motion Rate", *Control Theory and Applications*, Vol. 12, No. 2, pp. 138-46, Apr 1995.

43 Lucibello, P., and Benedetto, M., "Output Tracking for a Nonlinear Flexible Arm", *Transactions of ASME*, Vol. 115, pp. 78-85, Mar 1993.

44 Ly, U., "A Design Algorithm for Robust Low-Order Controller," Ph.D. Dissertation, Department of Aeronautics and Astronautics, Stanford University, Nov 1982.

45 Ly, U., "Sandy User's Guide", A.J. Controls, Inc., Jun 1993.

46 Ly, U., Van Steenwyk, B., and Schoemig, E., "Robust Control Design Using Parameter Optimization Techniques", *Journal of Control and Dynamic Systems*, Vol. 56, Academic Press, 1993.

47 Mandal, N., and Payandeh, S., "Control Strategies for Robotic Contact Tasks: An Experimental Study", *J. of Robotic Systems*, Vol. 12., No. 1, pp. 67-92, 1995.

48 Maples, J., "Force Control of Robotic Manipulators with Structural Flexibility", Ph.D. Dissertation, Dept. of Mechanical Engineering, Stanford University, Jun 1985.

49 Matsuno, F., and Yamamoto, K., "Dynamic Hybrid Position/Force Control of a Two Degree-of-Freedom Flexible Manipulator", *J of Robotic Systems*, Vol. 11, No. 5, pp.355-366, 1994

50 McLean, W., and Nelson, E., "Engineering Mechanics: Statics and Dynamics (4th Edition)", McGraw-Hill, New York, 1988.

51 Meressi, T. and Paden, B., "Gain Scheduled H_∞ Controllers for a Two Link Flexible Manipulator," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 3, pp. 537-543, 1994.

52 Moore, B., "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction", *IEEE Transactions on Automatic Control*, Vol. 26, No. 1, 1981.

53 Mills, J. and Lokhorst, D., "Experimental Results in Manipulator Contact Task Control", *Proc of the IEEE Int. Conf on Robotics and Automation*, pp. 1645-1651, 1991.

54 Nakajima, S., and Oyanagi, M., "Influence of Spatial Characteristics of Objects on a Surface Following Control System Equipped With an Ultrasonic Orientation Sensor", *Transactions of the Inst of Systems, Control, and Information Engineers*, Vol. 4, No. 3, pp. 126-32, Mar 1991.

55 Niemeyer G. and Slotine, J., "Performance in Adaptive Manipulator Control", P*roc of the 27th IEEE Conf on Decision and Control,* pp.1595-1591, Dec 1988.

56 Oakley, C., "Experiments in Modeling and End-Point Control of Two-Link Flexible Manipulators", Ph.D. Dissertation, Stanford University, 1991.

57 Oakley, C. and Cannon, R., "Anatomy of an Experimental Two-Link Flexible

Manipulator Under End-Point Control, *Proceedings of the IEEE Conf on Decision and Control*, Dec 1990.

58 Pham, C., Khalil, W., and Chevallereau, C., "A Nonlinear Model-Based Control of Flexible Robots", *Robotica*, Vol. 11, pp. 73-82, 1993.

59 Pudney, C., "Surface Modelling for Surface Following Robots", *Australian Computer Science Communications*, Vol. 16, No. 1, Pt. A, pp. 43-54, 1994.

60 Radin, B., and Gershon, D., "Fuzzy Compliance Control of Robotic Assembly Tasks", *Proc of the 3rd IEEE Conf on Fuzzy Systems*, Vol. 2, pp. 819-24, Jun 1994.

61 Raibert, M., and Craig, J., "Hybrid Position/Force Control of Manipulators", *J. of Dynamic Systems, Measurement, and Control*, 102, pp. 126-133, 1981.

62 Saberi, A., Chen, B., and Sannuti, P., "Loop Transfer Recovery: Analysis and Design", Springer Verlag, 1993.

63 Saberi, A. and Sannuti, P., "Observer Design for Loop Transfer Recovery and for Uncertain Dynamical Systems", *IEEE Trans. on Automatic Control*, Vol. 35, No. 8, pp. 878-897, 1990.

64 Sannuti, P. and Saberi, A., "Special Coordinate Basis for Multivariable Linear Systems - Finite and Infinite Zero Structure, Squaring Down and Decoupling", *Int J. of Control*, Vol. 45, No. 5, pp.1655-1704, 1987.

65 Schmitz, E., and Ramey, M., "A Simplified Dynamic Model for an Experimental Planar Manipulator With an Elastic Forearm", *Proceedings of the American Control Conference*, Jun 1989.

66 Sinha, A. and Peng, H., "Near-Minimum-Time Control of a Flexible Robot Arm via Linear Programming", *Int J. of Robotics and Automation*, Vol. 7, No. 4, pp. 152-160.

67 Whitney, D., "Force Feedback Control of Manipulator Fine Motion", *J. of Dyn Syst Meas Control Trans ASME*, Vol. 99, pp. 91-97, 1977.

Appendix A Arm Modeling Details

A.1 Assumptions

To simplify the mathematical models, several assumptions are made. First, the air cushions provide torsional support and provide a planar motion for the manipulator. Also, each joint is assumed frictionless with zero backlash (ideal pinned joint action). In addition, no frictional effects are included for the elbow and tip pads due to the air cushion. Lumped spring-mass-damper (LSMD) models assume linear spring and damper elements, and therefore are valid only for small deflections about a nominal link condition. From an electrical standpoint, the amplifiers are assumed to provide constant amplification.

A.2 Parameters

Table A.1 shows the TLA parameters required by the LSMD manipulator model. Table A.2 lists the parameters and their descriptions [56].

Table A.1 -TLA Manipulator Parameters

Location	Link 1	Link 2	Units
Hub	I_{h1}=4.1238x10^{-3}	I_{h2}=5.1099x10^{-4}	kg-m^2
Hub	m_{h1}=1.1618	m_{h2}=0.5745	kg
Hub	c_{h1}=5.5959x10^{-5}	c_{h2}=0.0236	m
Hub	o_{h1}=0.05842	o_{h2}=0.1143	m
Link	m_{L1}=0.1493	m_{L2}=0.4155	kg
Link	L_1=0.4699	L_2=0.4699	m
Link	E_1=70x10^9	E_2=70x10^9	N/m^2
Link	t_1=0.003175	t_2=0.003175	m
Link	h_1=0.0381	h_2=0.0381	m
Tip	I_{t1}=0.0058	I_{t2}=0.003	kg-m^2
Tip	m_{t1}=2.0755	m_{t2}=0.35	kg
Tip	c_{t1}=4.3807x10^{-4}	c_{t2}=4.3807x10^{-4}	m
Tip	o_{t1}=0.0063	o_{t2}=0.0063	m

For the two-link manipulator used in this dissertation, values for the physical parameters were experimentally determined [22]. Some values were measured with a precision ruler, while others were based on calculations from measurements. In addition, the elastic parameter k was determined from measurement of tip deflection from a known force. The damping parameter b was determined from an iterative adjustment to make the simulation output match the experimental output in terms of fundamental damped frequency and decay rate. The damping coefficient varies with the number of links, and the only method available to obtain the correct value is through experimental tuning. Also, all the damping characteristics were assumed to be equal at each joint of the LSMD model to make the mathematics easier.

Table A.2 - LSMD Parameters

Parameter	Description
n_i	number of elements per link
θ_i	base rotation of link i
q_{ij}	jth elem rotational deflection
x_{ep}	global x Cartesian tip position
y_{ep}	global y Cartesian tip position
del_x_{ep}	local x Cartesian tip position
del_y_{ep}	local x Cartesian tip position
o_{hi}	distance from shaft i to link i attachment to hub
o_{ti}	distance from tip attachment point of link i to shaft i+1
c_{hi}	distance from shaft i to the c.o.m of hub i
c_{ti}	distance from motor shaft i + 1 to c.o.m of tip i
I_i	length of element for link i
I_{hi}	Inertia of hub i about its c.o.m
I_{ti}	Inertia of tip i about its c.o.m
I_{li}	Inertia of link i elements about their c.o.m's
m_{hi}	hub i mass
m_{ti}	tip i mass
m_{li}	link i element masses
m_{Li}	total link mass
k_i	link i element torsional stiffness
b_i	link i element torsional damping
τ_i	motor torque at joint i

A.3 Nonlinear Equations of Motion

The nonlinear equations of motion developed by Oakley [56] used Kane's method [30]. It is convenient to express a generalized speed vector **u** as:

$$u_{i_1} = \dot{\theta}_1 + \sum_{k=1}^{i_1-1} \dot{q}_{1k} ; i_1 = 1, \ldots, n_1 \tag{A.1}$$

$$u_{n_1+i_2} = \dot{\theta}_1 + \sum_{k=1}^{n_1-1} \dot{q}_{1k} + \dot{\theta}_2 + \sum_{k=1}^{i_2-1} \dot{q}_{2k} ; i_2 = 1, \ldots, n_2 \tag{A.2}$$

$$u = [u_1, \ldots, u_{n_1}, u_{n_1+1}, \ldots, u_{n_1+n_2}]^T \tag{A.3}$$

where $\dot{\theta}_1$ is the shoulder motor velocity, $\dot{\theta}_2$ is the elbow motor velocity, $\dot{q}_{1k}$ is a sublink velocity for link 1 and $\dot{q}_{2k}$ is a sublink velocity for link 2. Additionally, the **z** vector contains the position of the manipulator and the τ vector contains the motor torque.

$$z = [\theta_1, q_{11,\ldots}, q_{1n_1}, \theta_2, q_{21}, \ldots, q_{2n_2}]^T \tag{A.4}$$

$$\tau = [\tau_1, \tau_2]^T \tag{A.5}$$

The complete non-linear matrix equation for a two-link manipulator is therefore represented by the following equation:

$$M(z)\dot{u} + V(z, u) + R\dot{z} + Kz = T\tau \tag{A.6}$$

where **M** possesses the mass and inertia properties, **V** contains the centripetal and Coriolis terms, **R** possesses the Coulomb torsional damping, and **K** contains the torsional stiffness. Also, **T** is the torque input distribution matrix.

A.4 Linearized Two-Link Equations of Motion

Linearization may occur about any joint condition with zero link deflections and all velocity states equal to zero. Application of a Taylor's series expansion about the non-linear equations of motion in Equation (A.6) produces the desired linearization. The resulting linearized equation of motion is:

$$\tilde{\underset{\sim}{M}}\Delta\ddot{\underset{\sim}{z}} + \underset{\sim}{R}\Delta\dot{\underset{\sim}{z}} + \underset{\sim}{K}\Delta\underset{\sim}{z} = \underset{\sim}{T}\Delta\underset{\sim}{\tau} \tag{A.7}$$

where $\tilde{M}$ is a Taylor expansion of **M** about the chosen linearization condition. **V** from the non-linear equation disappears since it is a function of velocity squared, and velocity is assumed small near the linearization state.

It is convenient to place the linearized equations of motion in the state-space form of the following equations.

$$\begin{bmatrix}\tilde{\boldsymbol{M}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\frac{d}{dt}\begin{bmatrix}\Delta\dot{\underset{\sim}{z}}\\ \Delta\underset{\sim}{z}\end{bmatrix} = \begin{bmatrix}-\boldsymbol{R} & -\boldsymbol{K}\\ \boldsymbol{I} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\Delta\dot{\underset{\sim}{z}}\\ \Delta\underset{\sim}{z}\end{bmatrix} + \begin{bmatrix}\boldsymbol{T}\\ \boldsymbol{0}\end{bmatrix}\Delta\underset{\sim}{\tau} \tag{A.8}$$

$$\frac{d}{dt}\begin{bmatrix}\Delta\dot{\underset{\sim}{z}}\\ \Delta\underset{\sim}{z}\end{bmatrix} = \begin{bmatrix}\tilde{\boldsymbol{M}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^{-1}\begin{bmatrix}-\boldsymbol{R} & -\boldsymbol{K}\\ \boldsymbol{I} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\Delta\dot{\underset{\sim}{z}}\\ \Delta\underset{\sim}{z}\end{bmatrix} + \begin{bmatrix}\tilde{\boldsymbol{M}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{T}\\ \boldsymbol{0}\end{bmatrix}\Delta\tau \tag{A.9}$$

$$\dot{\underset{\sim}{x}} = \boldsymbol{A}\underset{\sim}{x} + \boldsymbol{B}\Delta\underset{\sim}{\tau} \tag{A.10}$$

$$\underset{\sim}{y} = \boldsymbol{C}\underset{\sim}{x} + \boldsymbol{D}\Delta\underset{\sim}{\tau} \tag{A.11}$$

$$\boldsymbol{A} = \begin{bmatrix}\tilde{\boldsymbol{M}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^{-1}\begin{bmatrix}-\boldsymbol{R} & -\boldsymbol{K}\\ \boldsymbol{I} & \boldsymbol{0}\end{bmatrix} \tag{A.12}$$

$$\boldsymbol{B} = \begin{bmatrix}\tilde{\boldsymbol{M}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{T}\\ \boldsymbol{0}\end{bmatrix} \tag{A.13}$$

$$\underset{\sim}{x} = \begin{bmatrix}\Delta\dot{\underset{\sim}{z}}^{T} & \Delta\underset{\sim}{z}^{T}\end{bmatrix}^{T} \tag{A.14}$$

Matlab m-files tlapar.m and tlagen.m achieve the linearization using this scheme. Also, the m-files are converted to MATRIXx form in tlapar.mws and tlagen.mws to achieve the linearization. The tlapar.m file contains the parameters which are specific to the manipulator model, such as the hub inertia, as well as containing the number of elements per link desired. After loading the parameters, running the tlagen.m file actually generates the new A and B matrices, and calls them Abar and Bbar. The inputs to the model is the torque vector and the output contains the states as well as x and y endpoint positions. The original work for the m-files was done by Evers [22].

A.5 Non-Linear Two-Link 6 DOF Dynamic Equations

To provide a feel for the non-linear dynamics, this section provides the development of a 6 DOF dynamic model for the two-link manipulator [22]. Three elements per link provide the basis for a 6 DOF model. Using the basis of 3 identical elements for link 1 and 3 identical elements for link 2 results in the following mass center velocities:

$$v_{h1}^2 = c_{h1}^2 u_1^2 \tag{A.15}$$

$$v_{l_{11}}^2 = \left(o_{h1} + \frac{l1}{2}\right)^2 u_1^2 \tag{A.16}$$

$$v_{l_{12}}^2 = u_1^2 (o_{h1} + l1)^2 + u_2^2\left(\frac{l1}{2}\right)^2 + u_1 u_2 (o_{h1} + l1)\, l1 \cos(-q_{11}) \tag{A.17}$$

$$v_{l_{13}}^2 = u_1^2 (o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2\left(\frac{l1}{2}\right)^2 + \ldots \tag{A.18}$$

$$\ldots + 2u_1u_2 (o_{h1} + l1)\, l1 \cos(-q_{11}) + u_1u_3 (o_{h1} + l1)\, l1 \cos(-q_{11} - q_{12}) + \ldots$$

$$\ldots + u_2u_3 l1 \cos(-q_{12})$$

$$v_{t1}^2 = u_1^2 (o_{h1} + l1) + u_2^2 l1^2 + u_3^2 (l1 + o_{t1} + c_{t1})^2 + \ldots \tag{A.19}$$

$$\ldots + 2u_1u_3 (o_{h1} + l1)\, (l1 + o_{t1} + c_{t1}) \cos(-q_{11} - q_{12}) + \ldots$$

$$\ldots + 2u_1u_2 (o_{h1} + l1)\, l1 \cos(-q_{11}) + 2u_2u_3 l1\, (l1 + o_{t1} + c_{t1}) \cos(-q_{12})$$

$$v_{h2}^2 = u_1^2 (o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2 (l1 + o_{t1})^2 + u_4^2 c_{h2}^2 + \ldots \tag{A.20}$$

$$\ldots + 2u_1u_1 (o \ \ + l1)\, (l1 + o \ \ ) \cos(-q \ \ - q \ \ ) + \ldots$$

$$\ldots + 2u_1u_2 l1\, (l1 + o_{h1}) \cos(-q_{11}) + 2u_2u_3 l1\, (l1 + o_{t1}) \cos(-q_{12}) + \ldots$$

$$\ldots + 2u_1u_4 (o_{h1} + l1)\, c_{h2} \cos(-q_{11} - q_{12} - q_2) + \ldots$$

$$\ldots + 2u_2u_4 l1\, c_{h2} \cos(q_{12} - q_2) + 2u_3u_4 (l1 + o_{t1})\, c_{h2} \cos(-q_2)$$

$$v_{l_{21}}^2 = u_1^2 (o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2 (l1 + o_{t1})^2 + u_4^2\left(o_{h2} + \frac{l2}{2}\right)^2 + \ldots \tag{A.21}$$

$$\ldots + 2u_1u_2 (o_{h1} + l1)\, l1 \cos(-q_{11}) + 2u_2u_3 l1\, (l1 + o_{t1}) \cos(-q_{12}) + \ldots$$

$$\ldots + (2u_1u_3 (o_{h1} + l1)\, (l1 + o_{t1}) \cos(-q_{11} - q_{12})) + \ldots$$

$$\ldots + \left(2u_1u_4 (o_{h1} + l1)\left(o_{h2} + \frac{l2}{2}\right) \cos(-q_{11} - q_{12} - q_2)\right) + \ldots$$

$$\ldots + 2u_2u_4 l1\left(o_{h2} + \frac{l2}{2}\right) \cos(-q_{12} - q_2) + \ldots$$

$$\ldots + 2u_3u_4\,(l1 + o_{t1})\left(o_{h2} + \frac{l2}{2}\right)\cos\,(-q_2)$$

$$v_{l_{21}}^2 \;=\; u_1^2\,(o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2\,(l1 + o_{t1})^2 + u_4^2\,(o_{h2} + l2)^2 + \ldots \qquad (A.22)$$

$$\ldots + u_5^2\left(\frac{l2}{2}\right)^2 + 2u_1u_2\,(o_{h1} + l1)\,l1\cos\,(-q_{11}) + \ldots$$

$$\ldots + 2u_1u_3\,(o_{h1} + l1)\;(l1 + o_{t1})\cos\,(-q_{11} - q_{12}) + \ldots$$

$$\ldots + 2u_1u_4\,(o_{h1} + l1)\;(l2 + o_{h2})\cos\,(-q_{11} - q_{12} - q_2) + \ldots$$

$$\ldots + u_1u_5\,(o_{h1} + l1)\,l2\cos\,(-q_{11} - q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + 2u_2u_3 l1\,(o_{t1} + l1)\cos\,(-q_{12}) + \ldots$$

$$\ldots + 2u_2u_4 l1\,(o_{h2} + l2)\cos\,(-q_{12} - q_2) + \ldots$$

$$\ldots + u_2u_5 l1 l2\cos\,(-q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + 2u_3u_4\,(o_{t1} + l1)\;(l2 + o_{h2})\cos\,(-q_2) + \ldots$$

$$\ldots + u_3u_5\,(o_{t1} + l1)\,l2\cos\,(-q_2 - q_{21}) + u_4u_5\,(o_{h2} + l2)\,l2\cos\,(-q_{21})$$

$$v_{l_{23}}^2 \;=\; u_1^2\,(o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2\,(l1 + o_{t1})^2 + u_4^2\,(o_{h2} + l2)^2 + \ldots \qquad (A.23)$$

$$\ldots + u_5^2\,(l2)^2 + u_6^2\left(\frac{l2}{2}\right)^2 + 2u_1u_2\,(o_{h1} + l1)\,l1\cos\,(-q_{11}) + \ldots$$

$$\ldots + 2u_1u_3\,(o_{h1} + l1)\;(l1 + o_{t1})\cos\,(-q_{11} - q_{12}) + \ldots$$

$$\ldots + 2u_1u_4\,(o_{h1} + l1)\;(l2 + o_{h2})\cos\,(-q_{11} - q_{12} - q_2) + \ldots$$

$$\ldots + 2u_1u_5\,(o_{h1} + l1)\,l2\cos\,(-q_{11} - q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + u_1u_6\,(o_{h1} + l1)\,l2\cos\,(-q_{11} - q_{12} - q_2 - q_{21} - q_{22}) + \ldots$$

$$\ldots + 2u_2u_3 l1\,(o_{t1} + l1)\cos\,(-q_{12}) + 2u_2u_4 l1\,(o_{h2} + l2)\cos\,(-q_{12} - q_2) + \ldots$$

$$\ldots + 2u_2u_5 l1 l2\cos\,(-q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + u_2u_6 l1 l2\cos\,(-q_{12} - q_2 - q_{21} - q_{22}) + \ldots$$

$$\ldots + 2u_3u_4\,(o_{t1} + l1)\;(l2 + o_{h2})\cos\,(-q_2) + \ldots$$

$$\ldots + 2u_3u_5\,(o_{t1} + l1)\,l2\cos\,(-q_2 - q_{21}) + 2u_4u_5\,(o_{h2} + l2)\,l2\cos\,(-q_{21})$$

$$\ldots + u_3 u_6 (l1 + o_{t1})\, l2 \cos(-q_{22} - q_2 - q_{21}) + \ldots$$

$$\ldots + u_4 u_6 (o_{h2} + l2)\, l2 \cos(-q_{21} - q_{22}) + u_5 u_6 l2^2 \cos(-q_{22})$$

$$v_{t2}^2 = u_1^2 (o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2 (l1 + o_{t1})^2 + u_4^2 (o_{h2} + l2)^2 + \ldots \quad \text{(A.24)}$$

$$\ldots + u_5^2 (l2)^2 + u_6^2 (l2 + o_{t2} + c_{t2})^2 + 2u_1 u_2 (o_{h1} + l1)\, l1 \cos(-q_{11}) + \ldots$$

$$\ldots + 2u_1 u_3 (o_{h1} + l1)\ (l1 + o_{t1}) \cos(-q_{11} - q_{12}) + \ldots$$

$$\ldots + 2u_1 u_4 (o_{h1} + l1)\ (l2 + o_{h2}) \cos(-q_{11} - q_{12} - q_2) + \ldots$$

$$\ldots + 2u_1 u_5 (o_{h1} + l1)\, l2 \cos(-q_{11} - q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + 2u_1 u_6 (o_{h1} + l1)\ (l2 + o_{t2} + c_{t2}) \cos(-q_{11} - q_{12} - q_2 - q_{21} - q_{22}) + \ldots$$

$$\ldots + 2u_2 u_3 l1\, (o_{t1} + l1) \cos(-q_{12}) + 2u_2 u_4 l1\, (o_{h2} + l2) \cos(-q_{12} - q_2) + \ldots$$

$$\ldots + 2u_2 u_5 l1\, l2 \cos(-q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + 2u_2 u_6 l1\, (l2 + o_{t2} + c_{t2}) \cos(-q_{12} - q_2 - q_{21} - q_{22}) + \ldots$$

$$\ldots + 2u_3 u_4 (o_{t1} + l1)\ (l2 + o_{h2}) \cos(-q_2) + \ldots$$

$$\ldots + 2u_3 u_5 (o_{t1} + l1)\, l2 \cos(-q_2 - q_{21}) + 2u_4 u_5 (o_{h2} + l2)\, l2 \cos(-q_{21})$$

$$\ldots + 2u_3 u_6 (l1 + o_{t1})\ (l2 + o_{t2} + c_{t2}) \cos(-q_{22} - q_2 - q_{21}) + \ldots$$

$$\ldots + 2u_4 u_6 (o_{h2} + l2)\ (l2 + ot2 + ct2) \cos(-q_{21} - q_{22}) + \ldots$$

$$\ldots + 2u_5 u_6 l2\, (l2 + o_{t2} + c_{t2}) \cos(-q_{22})$$

The current equations are unwieldy, so generalized inertia terms are introduced. The Lagrangian development of the equations of motion to follow uses these generalized inertia terms.

$$J_{11} = m_{h1} c_{h1}^2 + I_{h1} + m_{l1}\left(o_{h1} + \frac{l1}{2}\right)^2 + I_{l1} + \ldots \quad \text{(A.25)}$$

$$\ldots + (2m_{11} + m_{t1} + m_{h2} + 3m_{l2} + m_{t2})\ (o_{h1} + l1)^2$$

$$J_{22} = m_{l1}\left(\frac{l1}{2}\right)^2 + I_{t1} + I_{l1} + (m_{l1} + m_{t1} + m_{h2} + 3m_{l2} + m_{t2})\, l1^2 \tag{A.26}$$

$$J_{33} = m_{l1}\left(\frac{l1}{2}\right)^2 + I_{l1} + m_{t1}\,(l1 + o_{t1} + c_{t1})^2 + \ldots \tag{A.27}$$

$$\ldots + (m_{h2} + m_{l2} + m_{t2})\,(l1 + o_{t1})^2$$

$$J_{44} = m_{h2}c_{h2}^2 + I_{h2} + m_{l2}\left(o_{h2} + \frac{l2}{2}\right)^2 + I_{l2} + (2m_{l2} + m_{t2})\,(o_{h2} + l2)^2 \tag{A.28}$$

$$J_{55} = m_{l2}\left(\frac{l2}{2}\right)^2 + I_{l2} + (m_{l2} + m_{t2})\, l2^2 \tag{A.29}$$

$$J_{66} = m_{l2}\left(\frac{l2}{2}\right)^2 + I_{l2} + m_{t2}\,(l2 + o_{t2} + c_{t2})^2 + I_{t2} \tag{A.30}$$

$$J_{12} = \frac{1}{2}[\,(3m_{l1} + 2m_{t1} + 2m_{h2} + 6m_{l2} + 2m_{t2})\,(o_{h1} + l1)\, l1] \tag{A.31}$$

$$J_{13} = \frac{1}{2}(m_{l1}l1 + 2m_{t1}\,(l1 + o_{t1} + c_{t1}))\,(o_{h1} + l1) + \ldots \tag{A.32}$$

$$\ldots + \frac{1}{2}(2m_{h2} + 6m_{l2} + 2m_{t2})\,(l1 + o_{t1})\,(o_{h1} + l1)$$

$$J_{14} = \frac{1}{2}\left(2m_{h2}c_{h2} + 2m_{l2}\left(o_{h2} + \frac{l2}{2}\right)\right)(o_{h1} + l1) + \ldots \tag{A.33}$$

$$\ldots + \frac{1}{2}(4m_{l2} + 2m_{t2})\,(o_{h2} + l2)\,(o_{h1} + l1)$$

$$J_{15} = \frac{1}{2}[\,(3m_{l2} + 2m_{t2})\,(o_{h1} + l1)\, l1] \tag{A.34}$$

$$J_{16} = \frac{1}{2}[\,(m_{l2}l2 + 2m_{t2}\,(l2 + o_{t2} + c_{t2}))\,(o_{h1} + l1)\,] \tag{A.35}$$

$$J_{23} = \frac{1}{2}(m_{l1}l1 + 2m_{t1}(l1 + o_{t1} + c_{t1}))l1 + \dots \tag{A.36}$$

$$\dots + \frac{1}{2}(2m_{h2} + 6m_{l2} + 2m_{t2})\ (l1 + o_{t1})l1$$

$$J_{24} = \frac{1}{2}\left(2m_{h2}c_{h2} + 2m_{l2}\left(o_{h2} + \frac{l2}{2}\right)\right)(l1) + \dots \tag{A.37}$$

$$\dots + \frac{1}{2}(4m_{l2} + 2m_{t2})\ (o_{h2} + l2)\ (l1)$$

$$J_{25} = \frac{1}{2}[\ (3m_{l2})\, l2\,(o_{h2} + l2) + 2m_{t2}l2]\, l1 \tag{A.38}$$

$$J_{26} = \frac{1}{2}[\ (m_{l2}l2 + 2m_{t2}(l2 + o_{t2} + c_{t2}))\ (l1)\ ] \tag{A.39}$$

$$J_{34} = \frac{1}{2}\left(2m_{h2}c_{h2} + 2m_{l2}\left(o_{h2} + \frac{l2}{2}\right)\right)(o_{t1} + l1) + \dots \tag{A.40}$$

$$\dots + \frac{1}{2}(4m_{l2} + 2m_{t2})\ (o_{h2} + l2)\ (o_{t1} + l1)$$

$$J_{35} = \frac{1}{2}[\ (3m_{l2} + 2m_{t2})\ (o_{h1} + l1)\, l2] \tag{A.41}$$

$$J_{36} = \frac{1}{2}[\ (m_{l2}l2 + 2m_{t2}(l2 + o_{t2} + c_{t2}))\ (o_{t1} + l1)\ ] \tag{A.42}$$

$$J_{45} = \frac{1}{2}[\ (3m_{l2} + 2m_{t2})\ (o_{h2} + l2)\, l2] \tag{A.43}$$

$$J_{46} = \frac{1}{2}[\ (m_{l2}l2 + 2m_{t2}(l2 + o_{t2} + c_{t2}))\ (o_{h2} + l2)\ ] \tag{A.44}$$

$$J_{56} = \frac{1}{2}[\ (m_{l2}l2 + 2m_{t2}(l2 + o_{t2} + c_{t2}))\ (l2)\ ] \tag{A.45}$$

The kinetic energy expression for the 6 DOF system can be expressed using the generalized inertia terms as follows:

$$\begin{aligned}T = {} & \frac{J_{11}}{2}\dot{u}_1^2 + \frac{J_{22}}{2}\dot{u}_2^2 + \frac{J_{33}}{2}\dot{u}_3^2 + \frac{J_{44}}{2}\dot{u}_4^2 + \frac{J_{55}}{2}\dot{u}_5^{22} + \frac{J_{66}}{2}\dot{u}_6^2 + \ldots \\ & \ldots + J_{12}\dot{u}_1\dot{u}_2\cos(u_1 - u_2) + J_{13}\dot{u}_1\dot{u}_3\cos(u_1 - u_3) + \ldots \\ & \ldots + J_{14}\dot{u}_1\dot{u}_4\cos(u_1 - u_4) + J_{15}\dot{u}_1\dot{u}_5\cos(u_1 - u_5) + \ldots \\ & \ldots + J_{16}\dot{u}_1\dot{u}_6\cos(u_1 - u_6) + J_{23}\dot{u}_2\dot{u}_3\cos(u_2 - u_3) + \ldots \\ & \ldots + J_{24}\dot{u}_2\dot{u}_4\cos(u_2 - u_4) + J_{25}\dot{u}_2\dot{u}_5\cos(u_2 - u_5) + \ldots \\ & \ldots + J_{26}\dot{u}_2\dot{u}_6\cos(u_2 - u_6) + J_{34}\dot{u}_3\dot{u}_4\cos(u_3 - u_4) + \ldots \\ & \ldots + J_{35}\dot{u}_3\dot{u}_5\cos(u_3 - u_5) + J_{36}\dot{u}_3\dot{u}_6\cos(u_3 - u_6) + \ldots \\ & \ldots + J_{45}\dot{u}_4\dot{u}_5\cos(u_4 - u_5) + J_{46}\dot{u}_4\dot{u}_6\cos(u_4 - u_6) + \ldots \\ & \ldots + J_{56}\dot{u}_5\dot{u}_6\cos(u_5 - u_6)\end{aligned} \tag{A.46}$$

The potential energy in the torsional springs is:

$$\begin{aligned}V = {} & \frac{1}{2}k_1(u_2 - u_1)^2 + \frac{1}{2}k_1(u_3 - u_2)^2 + \frac{1}{2}k_2(u_5 - u_4)^2 + \ldots \\ & \ldots + \frac{1}{2}k_2(u_6 - u_5)^2\end{aligned} \tag{A.47}$$

Lagrange's expression in the following equation develops the equations of motion. Here, Q_i' are the generalized forces due to torsional damping.

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{u}_i}\right) - \frac{\partial T}{\partial u_i} = \frac{\partial V}{\partial u_i} + Q_i' \tag{A.48}$$

Applying the above equation for the 6 DOF system yields the equations of motion for the system as follows.

$$J_{11}\dot{u}_1 + J_{12}\dot{u}_2\cos(-q_{11}) + J_{13}\dot{u}_3\cos(-q_{11} - q_{12}) + \ldots \tag{A.49}$$

$$\ldots + J_{14}\dot{u}_4 \cos(-q_{11} - q_{12} - q_2) + J_{15}\dot{u}_5 \cos(-q_{11} - q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + J_{16}\dot{u}_6 \cos(-q_{11} - q_{12} - q_2 - q_{21} - q_{22}) + \ldots$$

$$\ldots + J_{12}u_2^2 \sin(-q_{11}) + J_{13}u_3^2 \sin(-q_{11} - q_{12}) + \ldots$$

$$\ldots + J_{14}u_4^2 \sin(-q_{11} - q_{12} - q_2) + J_{15}u_5^2 \sin(-q_{11} - q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + J_{16}u_6^2 \sin(-q_{11} - q_{12} - q_2 - q_{21} - q_{22}) = \ldots$$

$$\ldots = k_1 q_{11} + b_1(u_2 - u_1) + T_1$$

$$J_{22}\dot{u}_2 + J_{21}\dot{u}_1 \cos(-q_{11}) + J_{23}\dot{u}_3 \cos(-q_{12}) + \ldots \qquad \text{(A.50)}$$

$$\ldots + J_{24}\dot{u}_4 \cos(-q_{12} - q_2) + J_{25}\dot{u}_5 \cos(-q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + J_{26}\dot{u}_6 \cos(-q_{12} - q_2 - q_{21} - q_{22}) + \ldots$$

$$\ldots - J_{21}u_2^2 \sin(-q_{11}) + J_{23}u_3^2 \sin(-q_{12}) + \ldots$$

$$\ldots + J_{24}u_4^2 \sin(-q_{12} - q_2) + J_{25}u_5^2 \sin(-q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots + J_{26}u_6^2 \sin(-q_{12} - q_2 - q_{21} - q_{22}) = \ldots$$

$$\ldots = -k_1 q_{11} + k_1 q_{12} - b_1(u_2 - u_1) + b_1(u_3 - u_2)$$

$$J_{33}\dot{u}_3 + J_{31}\dot{u}_1 \cos(-q_{11} - q_{12}) + J_{32}\dot{u}_2 \cos(-q_{12}) + \ldots \qquad \text{(A.51)}$$

$$\ldots + J_{34}\dot{u}_4 \cos(-q_2) + J_{35}\dot{u}_5 \cos(-q_2 - q_{21}) + \ldots$$

$$\ldots + J_{36}\dot{u}_6 \cos(-q_2 - q_{21} - q_{22}) + \ldots$$

$$\ldots - J_{31}u_1^2 \sin(-q_{11} - q_{22}) - J_{32}u_2^2 \sin(-q_{12}) + \ldots$$

$$\ldots + J_{34}u_4^2 \sin(-q_2) + J_{35}u_5^2 \sin(-q_2 - q_{21}) + \ldots$$

$$\ldots + J_{36}u_6^2 \sin(-q_2 - q_{21} - q_{22}) = \ldots$$

$$\ldots = -k_1 q_{12} - b_1 (u_3 - u_2) - T_2$$

$$J_{44}\ddot{u}_4 + J_{41}\ddot{u}_1 \cos(-q_2 - q_{11} - q_{12}) + J_{42}\ddot{u}_2 \cos(-q_2 - q_{12}) + \ldots \quad \text{(A.52)}$$

$$\ldots + J_{43}\ddot{u}_3 \cos(-q_2) + J_{45}\ddot{u}_5 \cos(-q_{21}) + \ldots$$

$$\ldots + J_{46}\ddot{u}_6 \cos(-q_{21} - q_{22}) + \ldots$$

$$\ldots - J_{41}u_1^2 \sin(-q_{11} - q_{22} - q_2) - J_{42}u_2^2 \sin(-q_{12} - q_2) + \ldots$$

$$\ldots - J_{43}u_3^2 \sin(-q_2) + J_{45}u_5^2 \sin(-q_{21}) + \ldots$$

$$\ldots + J_{46}u_6^2 \sin(-q_{21} - q_{22}) = \ldots$$

$$\ldots = k_2 q_{12} + (b_2)(u_5 - u_4) + T_2$$

$$J_{55}\ddot{u}_5 + J_{51}\ddot{u}_1 \cos(-q_{11} - q_2 - q_{21} - q_{12}) + \ldots \quad \text{(A.53)}$$

$$\ldots + J_{52}\ddot{u}_2 \cos(-q_2 - q_{12} - q_{21}) + J_{53}\ddot{u}_3 \cos(-q_{21} - q_2) + \ldots$$

$$\ldots + J_{54}\ddot{u}_4 \cos(-q_{21}) + J_{56}\ddot{u}_6 \cos(-q_{22}) + \ldots$$

$$\ldots - J_{51}u_1^2 \sin(-q_{11} - q_{22} - q_2 - q_{21}) - J_{52}u_2^2 \sin(-q_{12} - q_2 - q_{21}) + \ldots$$

$$\ldots - J_{53}u_3^2 \sin(-q_2 - q_{21}) - J_{54}u_4^2 \sin(-q_{21}) + \ldots$$

$$\ldots + J_{56}u_6^2 \sin(-q_{22}) = \ldots$$

$$\ldots = -k_2 q_{12} + k_2 q_{12} - b_2 (u_5 - u_4) + b_2 (u_6 - u_5)$$

$$J_6\ddot{u}_6 + J_{61}\ddot{u}_1 \cos(-q_{11} - q_2 - q_{21} - q_{12} - q_{22}) + \ldots \quad \text{(A.54)}$$

$$\ldots + J_{62}\ddot{u}_2 \cos(-q_{21} - q_{22} - q_2 - q_{12}) + J_{63}\ddot{u}_3 \cos(-q_{21} - q_{22} - q_2) + \ldots$$

$$\ldots + J_{64}\ddot{u}_4 \cos(-q_{21} - q_{22}) + J_{65}\ddot{u}_5 \cos(-q_{22}) + \ldots$$

$$\ldots - J_{61}u_1^2 \sin(-q_{11} - q_{12} - q_{22} - q_2 - q_{21}) + \ldots$$

$$\ldots - J_{62}u_2^2\sin(-q_{12} - q_2 - q_{21} - q_{22}) - J_{63}u_3^2\sin(-q_2 - q_{21} - q_{22}) + \ldots$$

$$\ldots - J_{64}u_4^2\sin(-q_{21} - q_{22}) - J_{65}u_5^2\sin(-q_{22}) = \ldots$$

$$\ldots = -k_2 q_{22} + -b_2(u_6 - u_5)$$

A.6 Closed-Form Linearized Equations of Motion

The complete closed-form set of matrices for the linearized model are found in Oakley's work. The M on p. 259 of the dissertation is equivalent to (for the 6 DOF model):

$$\tilde{\boldsymbol{M}} = \begin{bmatrix} \tilde{\boldsymbol{M}}_{11} & \tilde{\boldsymbol{M}}_{12} \\ \tilde{\boldsymbol{M}}_{12} & \tilde{\boldsymbol{M}}_{22} \end{bmatrix} \tag{A.55}$$

where

$$\tilde{\boldsymbol{M}}_{11} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \tag{A.56}$$

$$\tilde{\boldsymbol{M}}_{12} = \begin{bmatrix} J_{14} & J_{15} & J_{16} \\ J_{24} & J_{25} & J_{26} \\ J_{34} & J_{35} & J_{36} \end{bmatrix} \cos(q_{2_{nom}}) \tag{A.57}$$

$$\tilde{\boldsymbol{M}}_{22} = \begin{bmatrix} J_{44} & J_{45} & J_{46} \\ J_{54} & J_{55} & J_{56} \\ J_{64} & J_{65} & J_{66} \end{bmatrix} \tag{A.58}$$

The terms were checked and were found to be in agreement with the exception of two typographical errors [22]. The corrections to p.260 of the dissertation are as follows:

$$M_{link1diag11} = \begin{pmatrix} e11 & 0 & \dots & \dots & \dots & 0 \\ 0 & e22 & 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 0 & \dots \\ \dots & \dots & \dots & 0 & en1 & 0 \\ 0 & \dots & \dots & \dots & 0 & enn \end{pmatrix} \tag{A.59}$$

where

$$e11 = m_{11}\left(o_{h1} + \frac{1}{2}l1\right)^2 + \sum_{i_1 = 2}^{n_1} m_{1i_1}(o_{h1} + l1)^2 \tag{A.60}$$

$$e22 = \frac{1}{4}m_{12}l1^2 + \sum_{i_1 = 3}^{n_1} m_{1i_1}l1^2 \tag{A.61}$$

$$en1 = \frac{1}{4}m_{1(n_1 - 1)}l1^2 + m_{1n_1}l1^2 \tag{A.62}$$

$$enn = \frac{1}{4}m_{1n_1}l1^2 \tag{A.63}$$

The second typographical error on p.260 should read:

$$M_{link2diag22} = \begin{pmatrix} e11 & 0 & \dots & \dots & \dots & 0 \\ 0 & e22 & 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 0 & \dots \\ \dots & \dots & \dots & 0 & en1 & 0 \\ 0 & \dots & \dots & \dots & 0 & enn \end{pmatrix} \tag{A.64}$$

where

$$e11 = m_{21}\left(o_{h2} + \frac{1}{2}l2\right)^2 + \sum_{i_2 = 2}^{n_2} m_{2i_2}\,(o_{h2} + l2)^2 \tag{A.65}$$

$$e22 = \frac{1}{4}m_{22}l2^2 + \sum_{i_2 = 3}^{n_2} m_{1i_2}l2^2 \tag{A.66}$$

$$en1 = \frac{1}{4}m_{2(n_2-1)}l2^2 + m_{1n_2}l2^2 \tag{A.67}$$

$$enn = \frac{1}{4}m_{2n_2}l2^2 \tag{A.68}$$

Appendix B - Additional SANDY Double Angle and PID Plots

This appendix provides a further reference for the SANDY double angle and PID designs. Specifically, the plots for trajectories 1 and 3, described in Chapter 5, are displayed for the PID controllers. Next, the SANDY controller for trajectories 1 and 3 are displayed. Finally, PID and SANDY are compared for trajectories 1 and 3. In all the plots with volts listed, the conversion for τ_1 volts to Nm is 1/1.9208 and the conversion for τ_2 volts to Nm is 1/8.045. Also, all the angle readings are relative to the starting positions. The choppiness on the angle plots is due to optical encoder quantization.

B.1 PID Plots

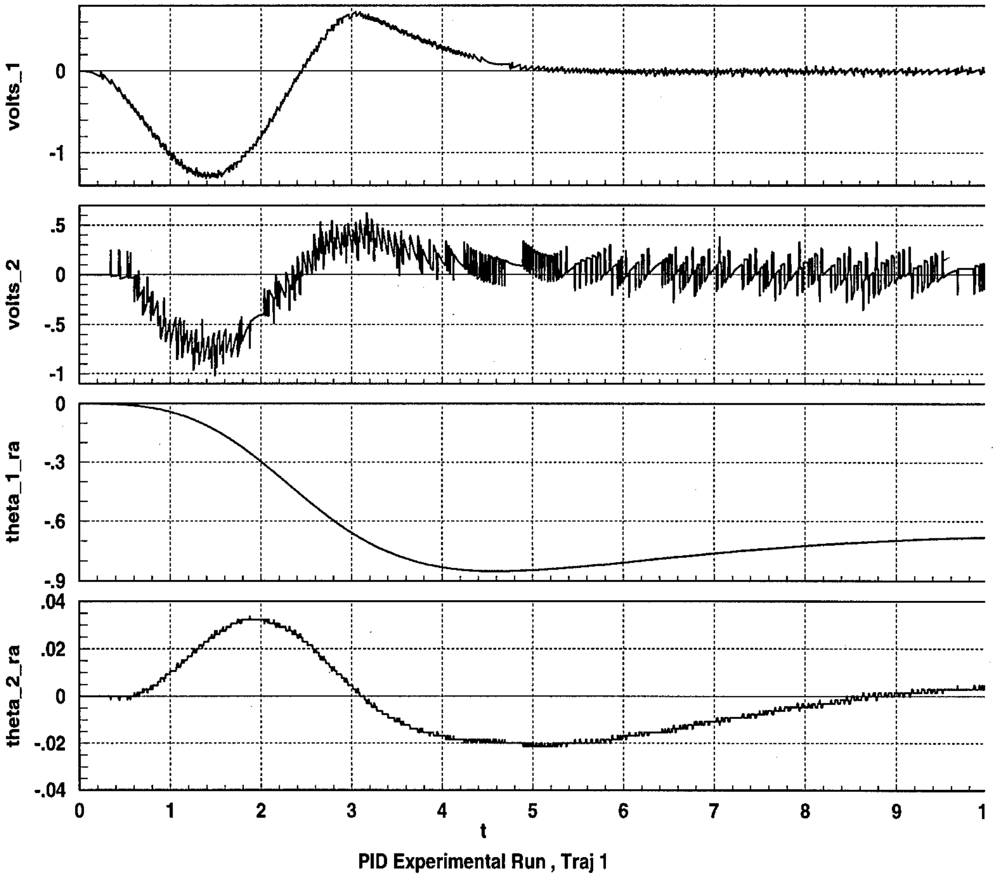


Figure B.1- PID Experimental Run, Trajectory 1

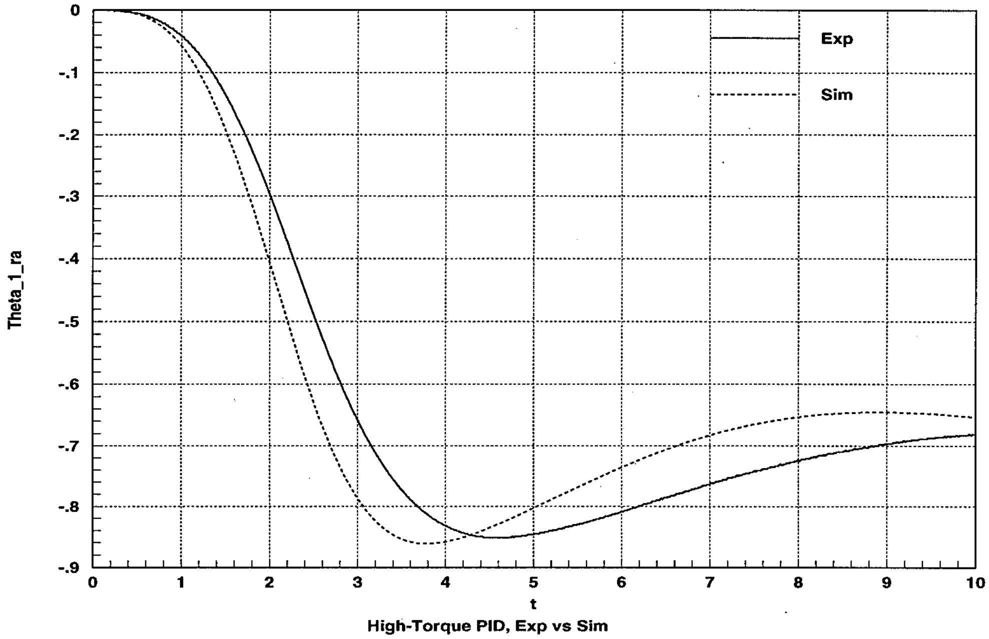


Figure B.2- PID Experimental vs. Simulation, Trajectory 1, θ_1

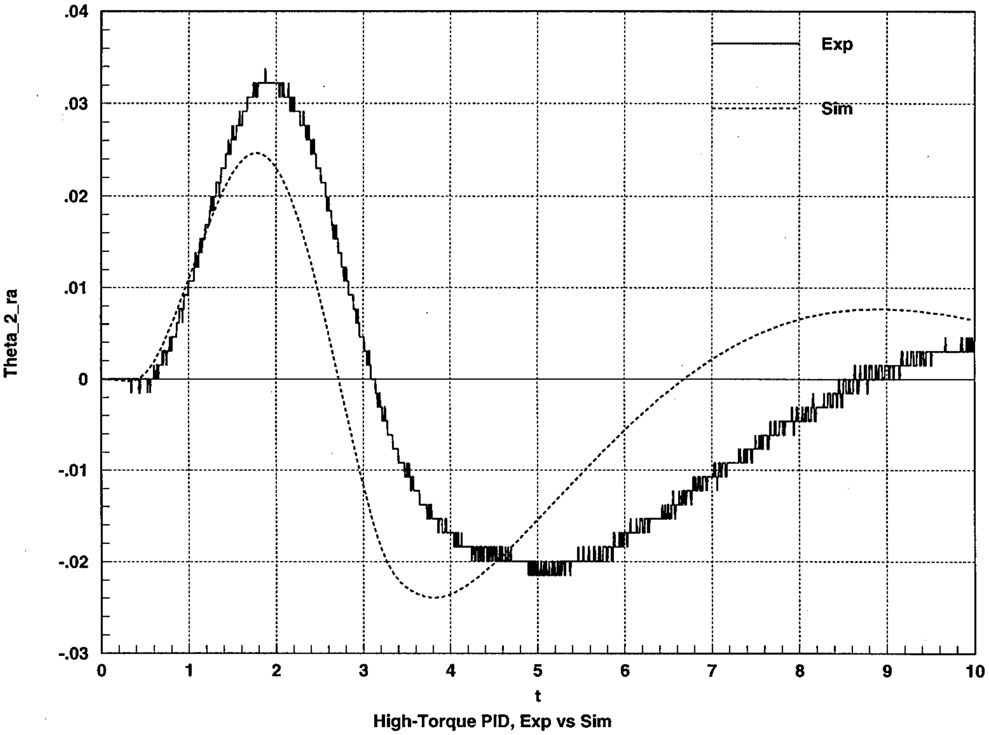


Figure B.3- PID Experimental vs. Simulation, Trajectory 1, θ_2

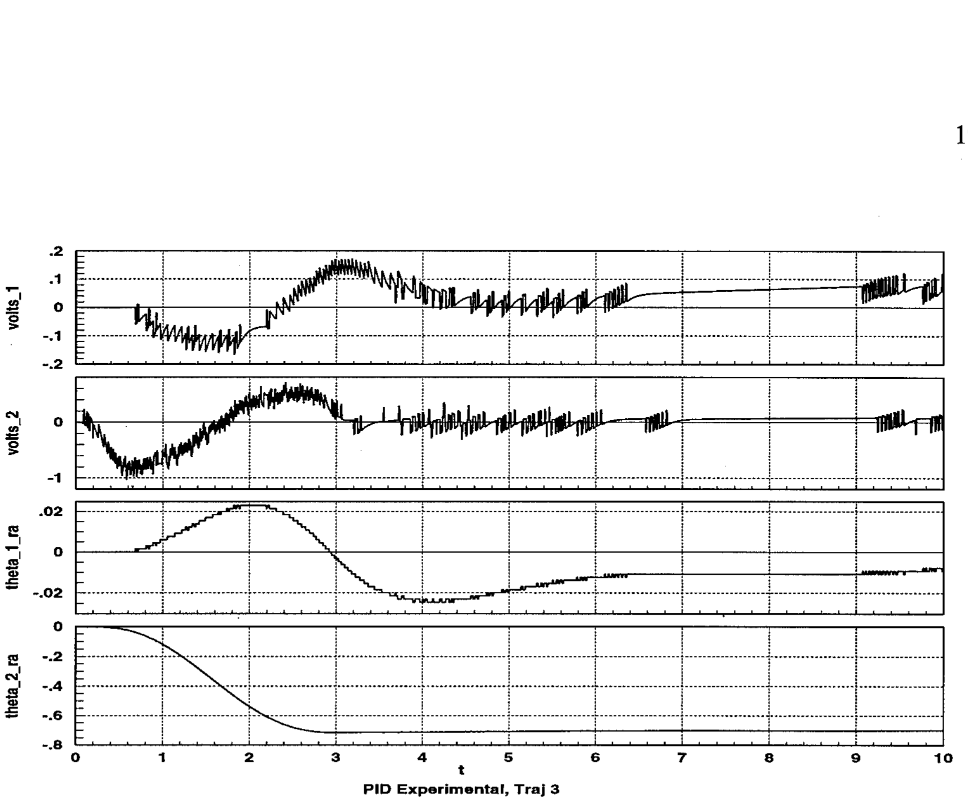


Figure B.4- PID Experimental Run, Trajectory 3

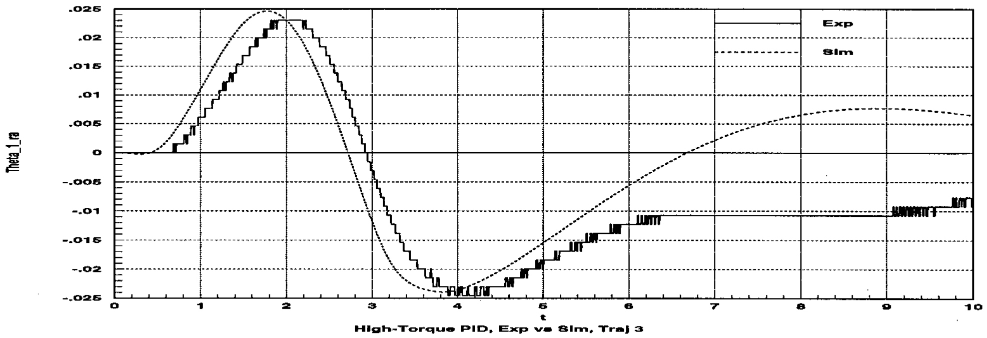


Figure B.5- PID Experimental vs. Simulation, Trajectory 3, θ_1

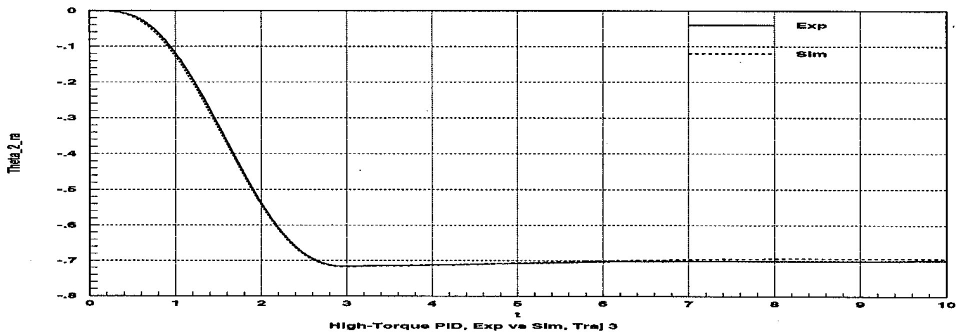


Figure B.6- PID Experimental vs. Simulation, Trajectory 3, θ_2

B.2 SANDY Double Angle Plots

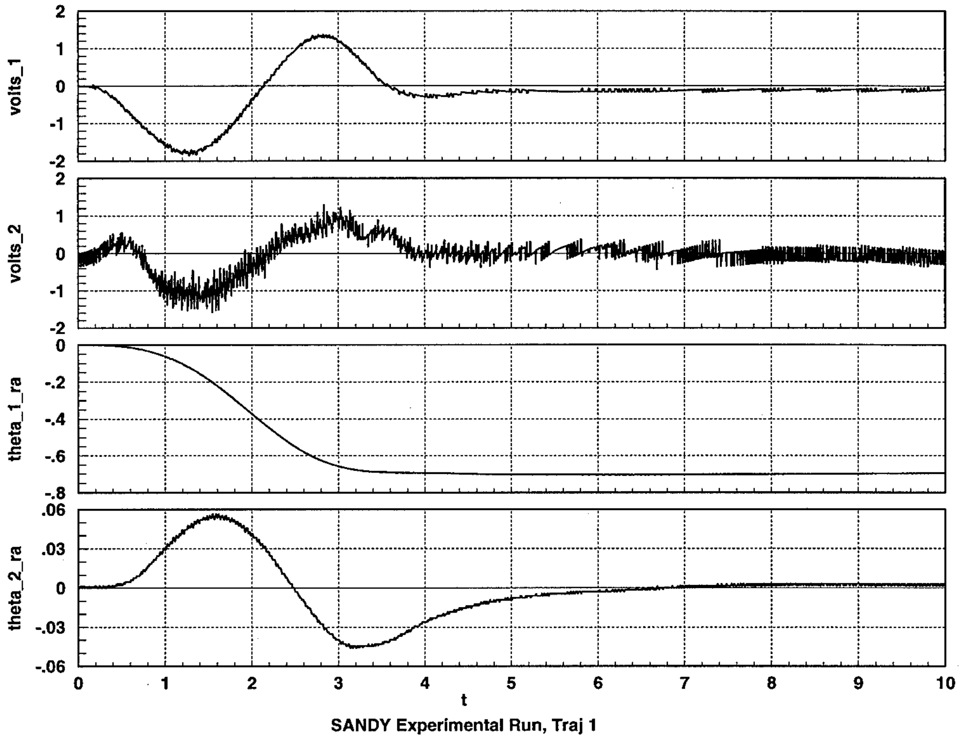


Figure B.7- SANDY Experimental Run, Trajectory 1

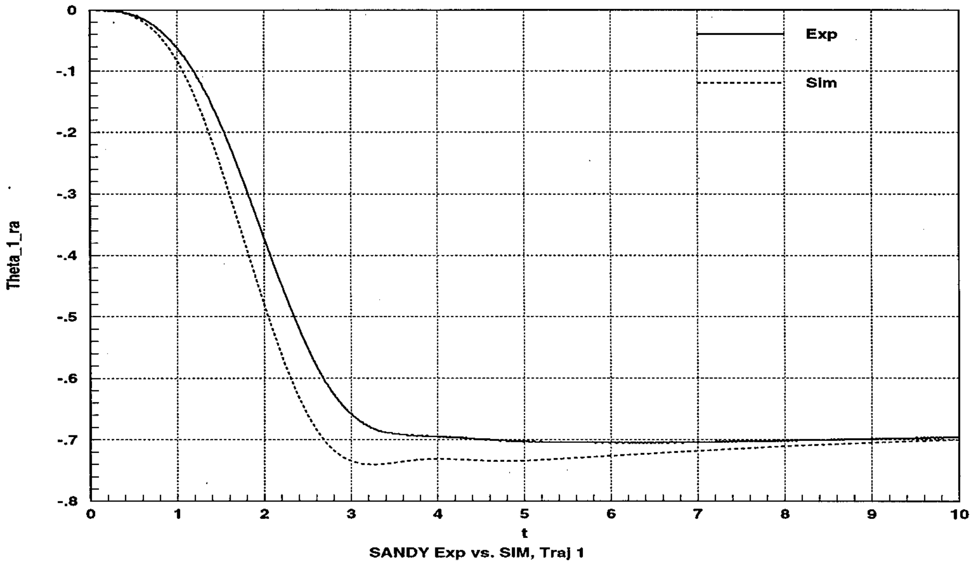


Figure B.8- SANDY Experimental vs. Simulation, Trajectory 1, θ_1

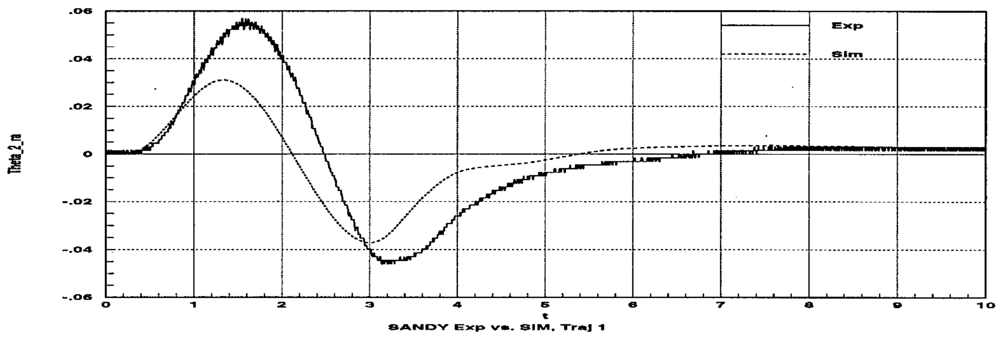


Figure B.9- SANDY Experimental vs. Simulation, Trajectory 1, θ_2

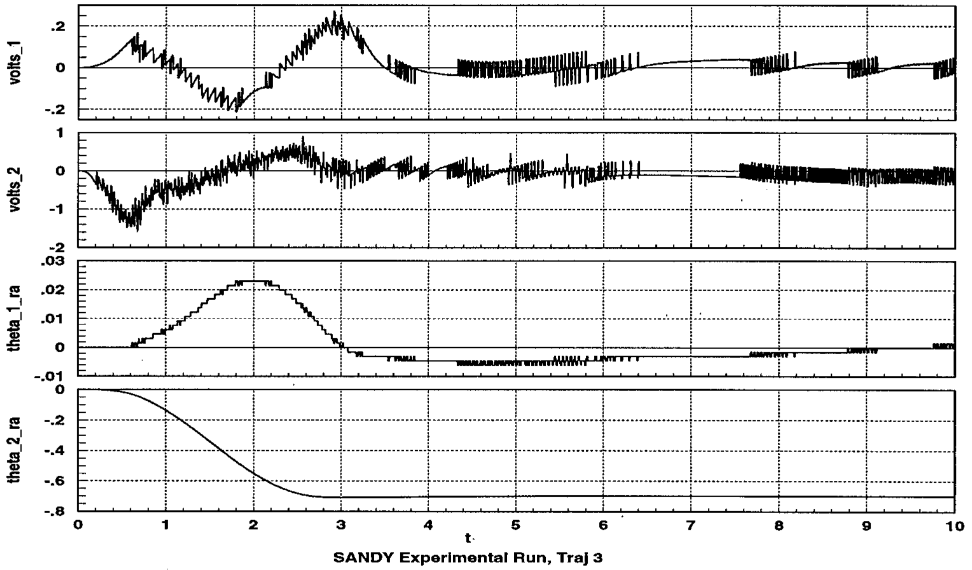


Figure B.10- SANDY Experimental Run, Trajectory 3

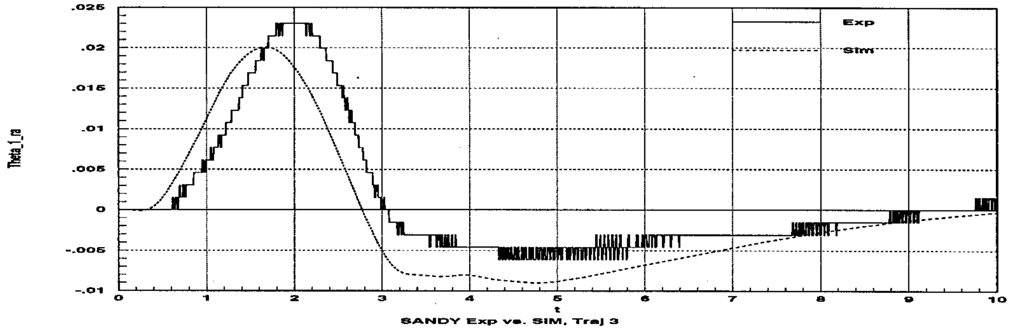


Figure B.11- SANDY Experimental vs. Simulation, Trajectory 3, θ_1

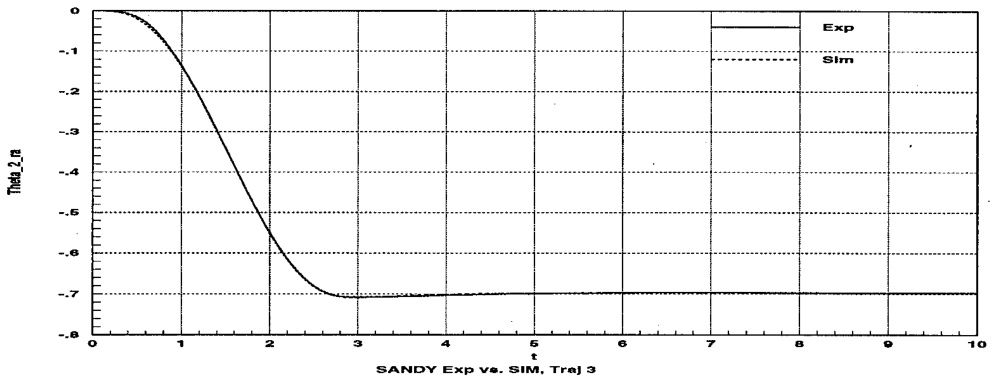


Figure B.12- SANDY Experimental vs. Simulation, Trajectory 3, θ_2

B.3 SANDY vs. PID Plots

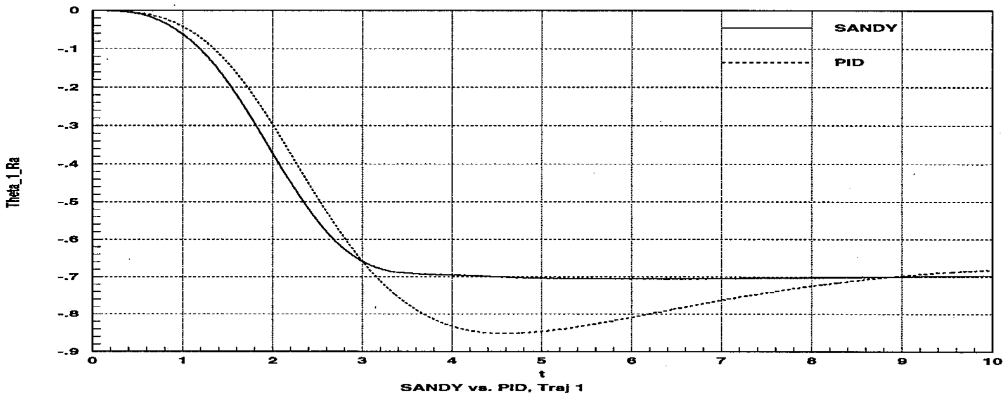


Figure B.13- SANDY vs. PID Experimental, Trajectory 1, θ_1

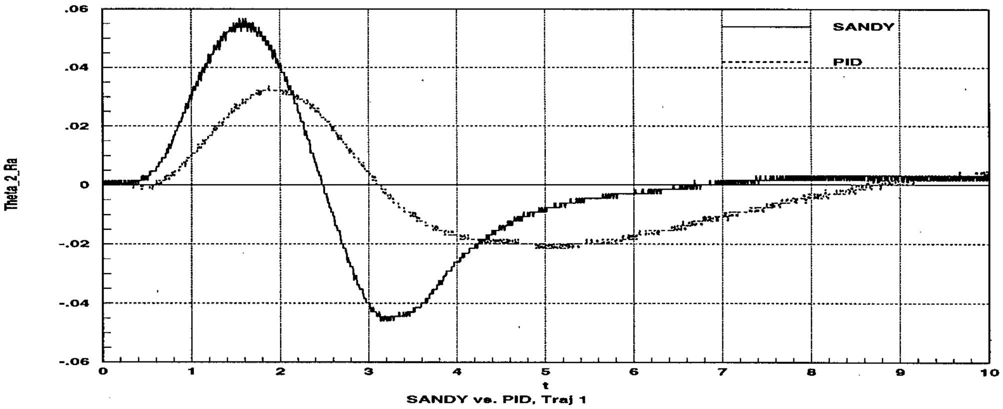


Figure B.14-SANDY vs. PID Experimental, Trajectory 1, θ_2

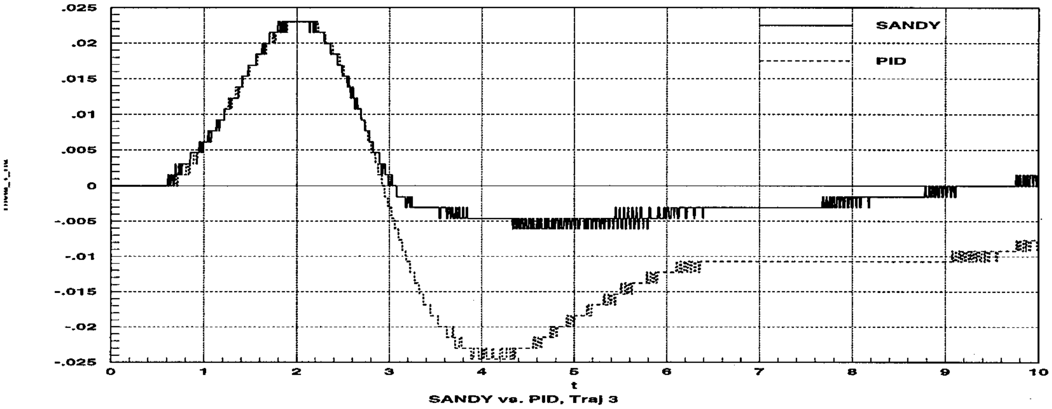


Figure B.15- SANDY vs. PID Experimental, Trajectory 3, θ_1

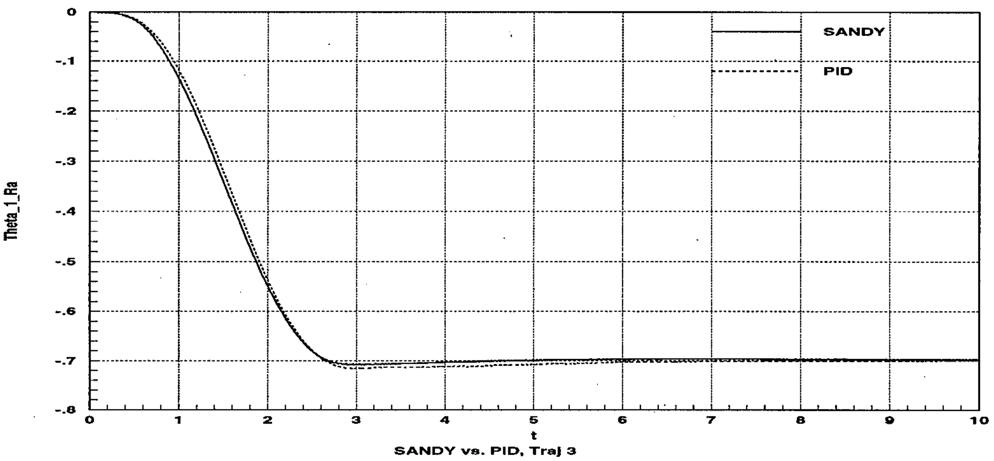


Figure B.16- SANDY vs. PID Experimental, Trajectory 3, θ_2

Appendix C - Additional LQG-Class Plots

This appendix provides a further reference for the LQG-class controllers. Those displayed here are LQG/LTR, LTR, 7th-order LQG/LTR, and 7th order LTR controllers. Specifically, the plots for trajectories 1 and 3, described in Chapter 5, are displayed for the LQG/LTR controller. Next, the LTR controller response for trajectories 1 and 3 are displayed. Then the 7th-order controller results are listed. Finally, PID and SANDY are compared with all the LQG-class controllers for trajectories 1 and 3. In all the plots with volts listed, the conversion for τ_1 volts to Nm is 1/1.9208 and the conversion for τ_2 volts to Nm is 1/8.045. The choppiness on some of the angle plots is due to quantization of the angle encoders.

C.1 LQG/LTR Plots

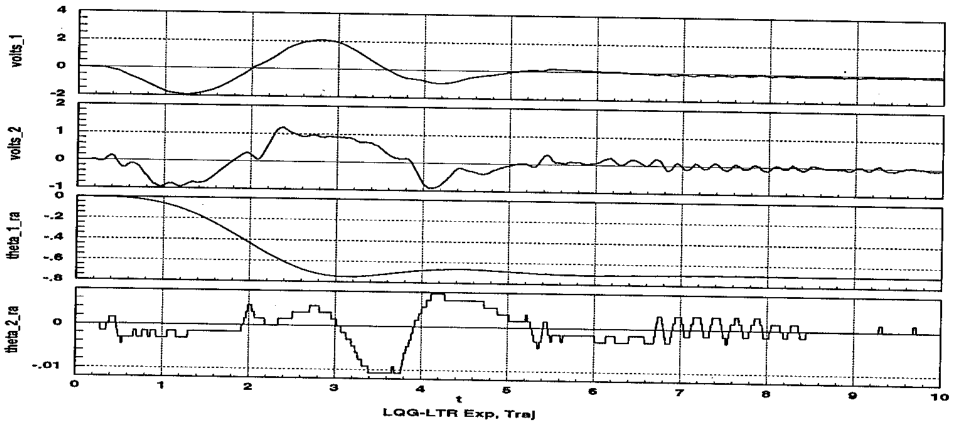


Figure C.1- LQG/LTR Experimental Run, Trajectory 1

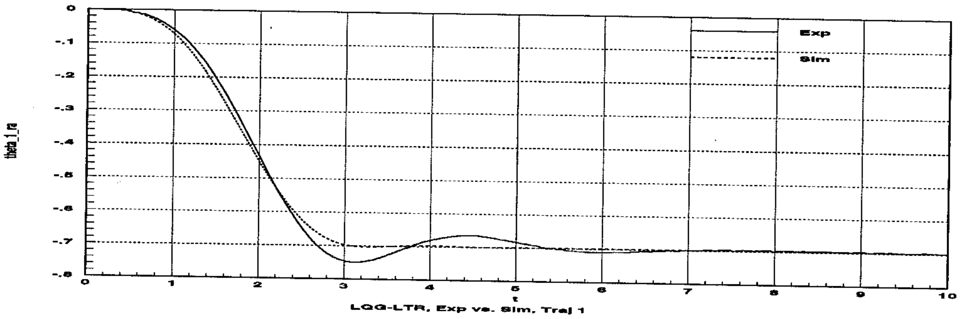


Figure C.2- LQG/LTR Experimental vs. Simulation, Trajectory 1, θ_1

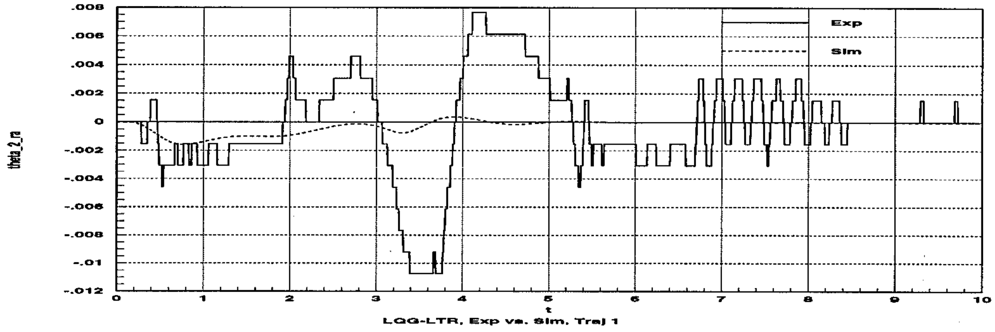


Figure C.3- LQG/LTR Experimental vs. Simulation, Trajectory 1, θ_2

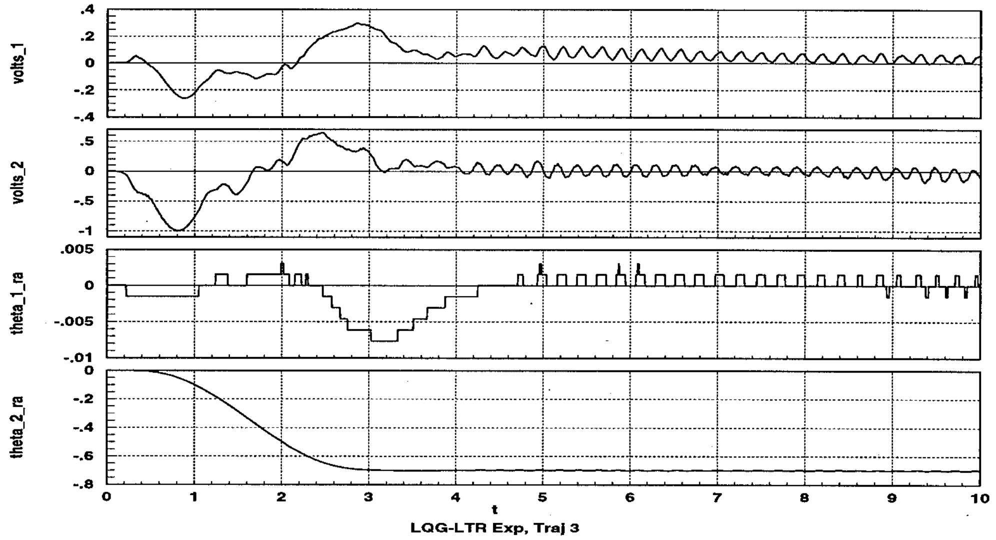


Figure C.4- LQG/LTR Experimental Run, Trajectory 3

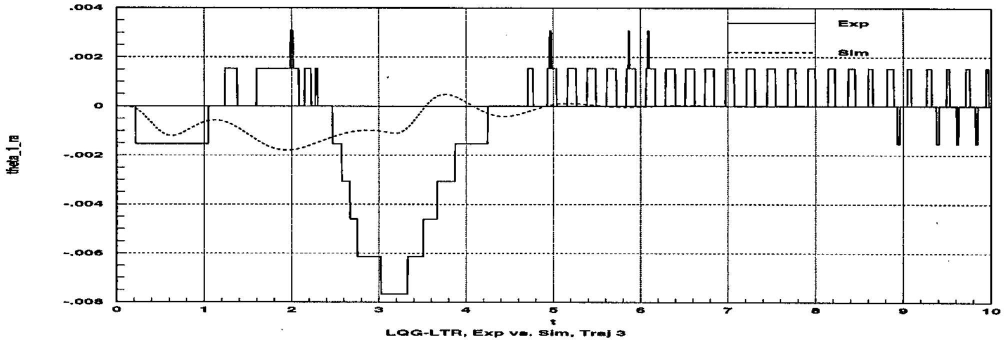


Figure C.5- LQG/LTR Experimental vs. Simulation, Trajectory 3, θ_1

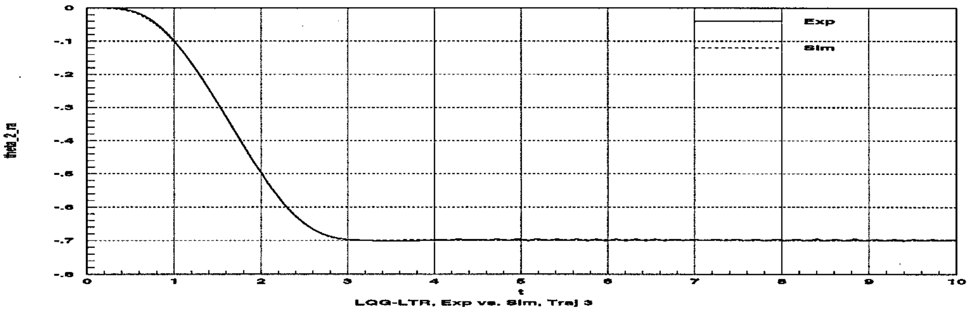


Figure C.6- LQG/LTR Experimental vs. Simulation, Trajectory 3, θ_2

C.2 LTR Plots

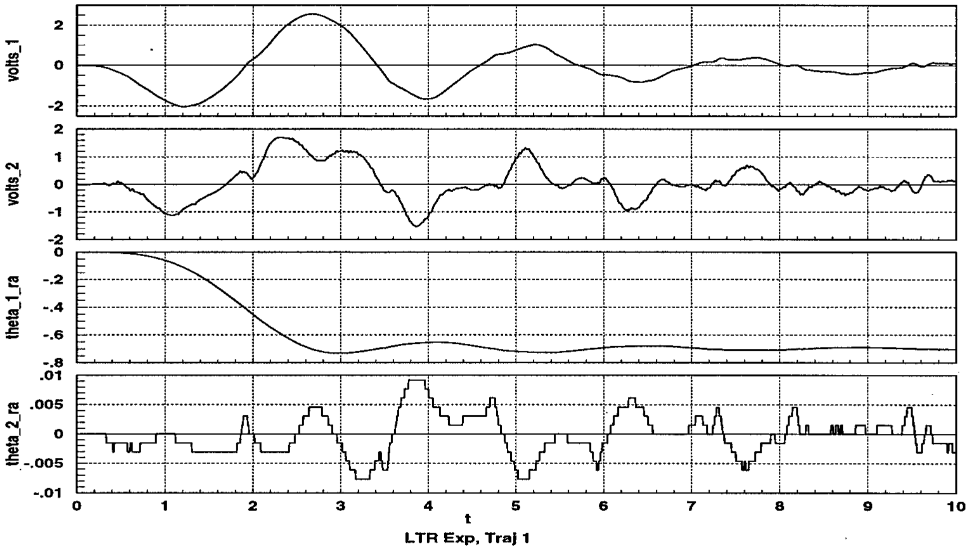


Figure C.7- LTR Experimental Run, Trajectory 1

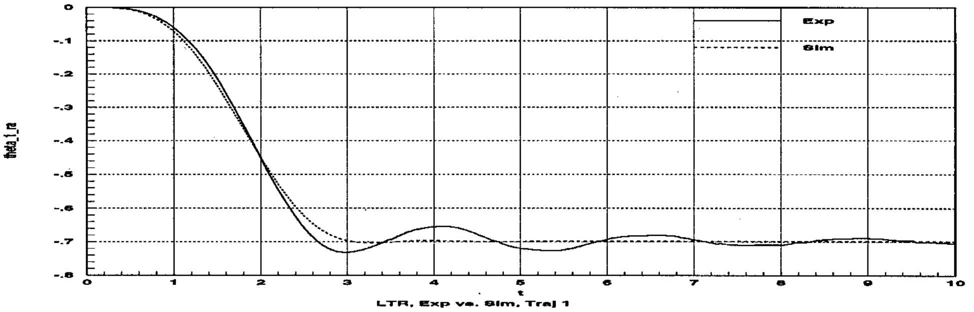


Figure C.8- LTR Experimental vs. Simulation, Trajectory 1, θ_1

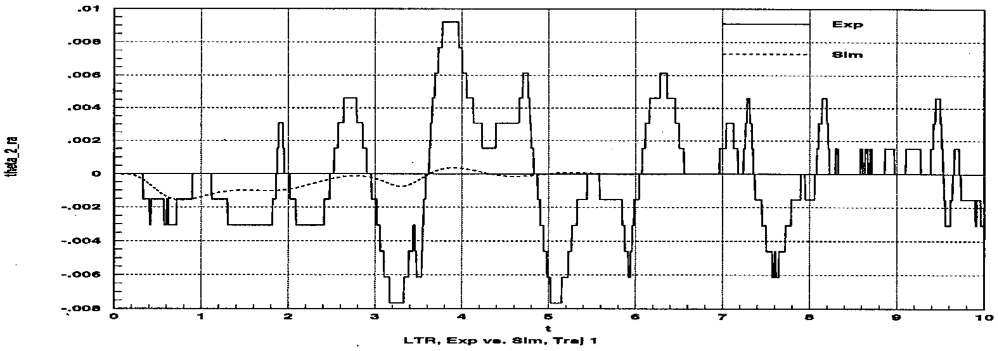


Figure C.9- LTR Experimental vs. Simulation, Trajectory 1, θ_2

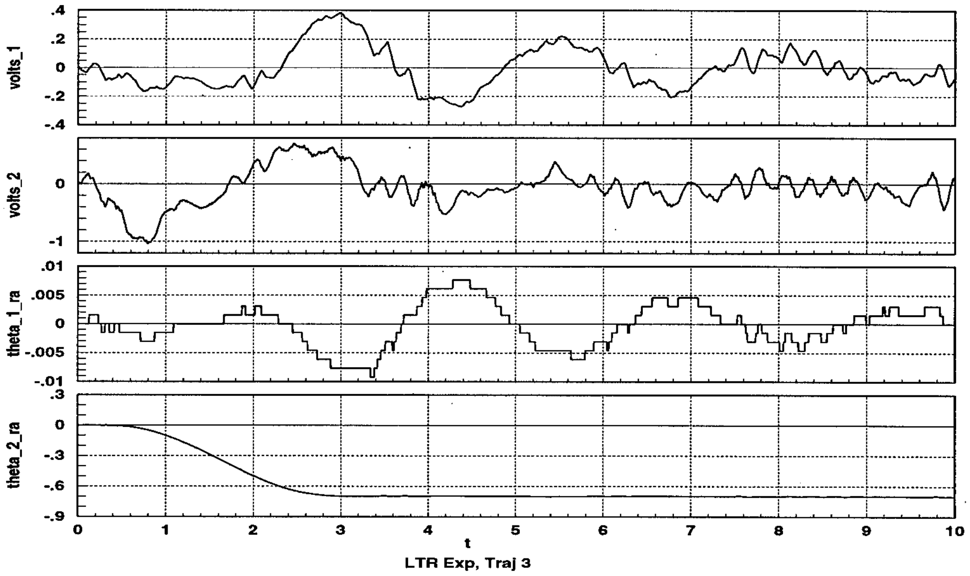


Figure C.10- LTR Experimental Run, Trajectory 3

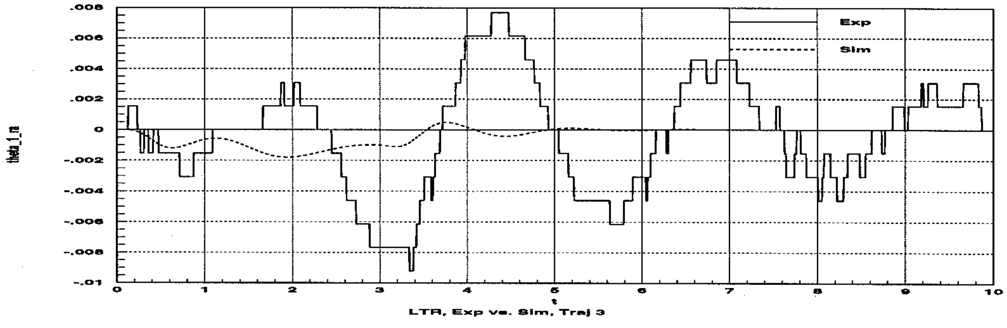


Figure C.11- LTR Experimental vs. Simulation, Trajectory 3, θ_1

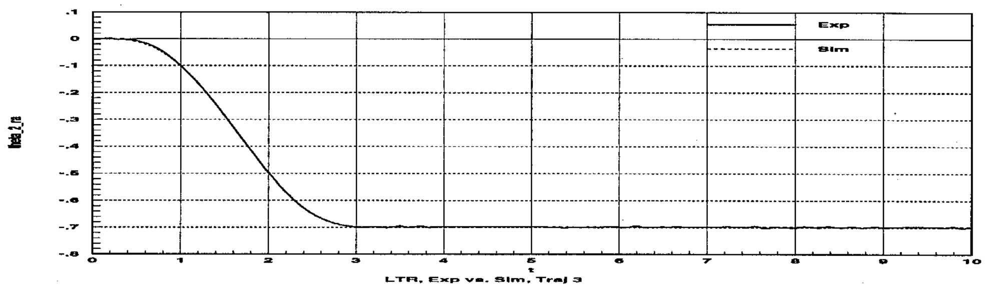


Figure C.12- LTR Experimental vs. Simulation, Trajectory 3, θ_2

C.3 7th Order LQG/LTR Plots

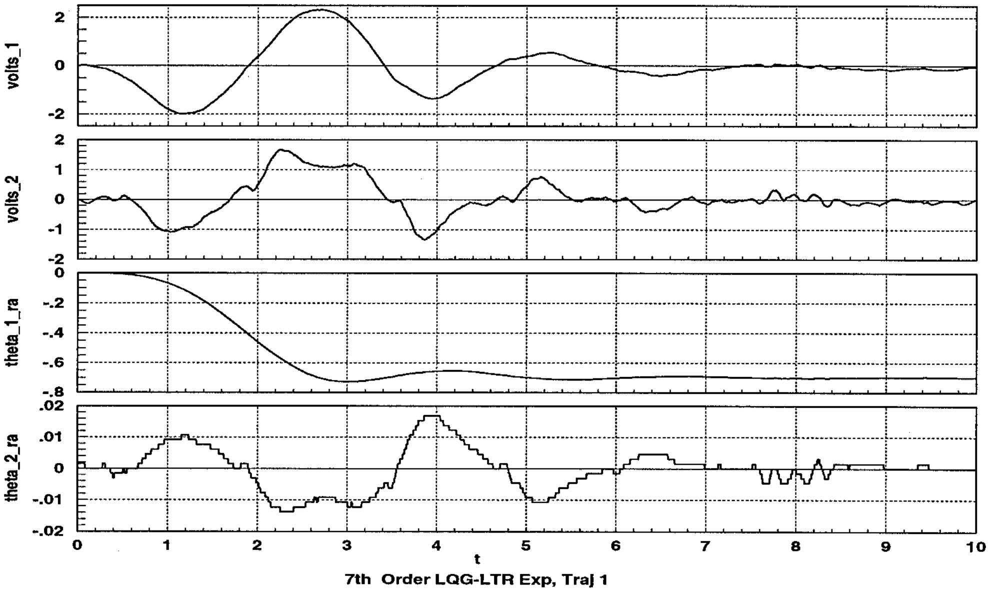


Figure C.13-7th Order LQG/LTR Experimental Run, Trajectory 1

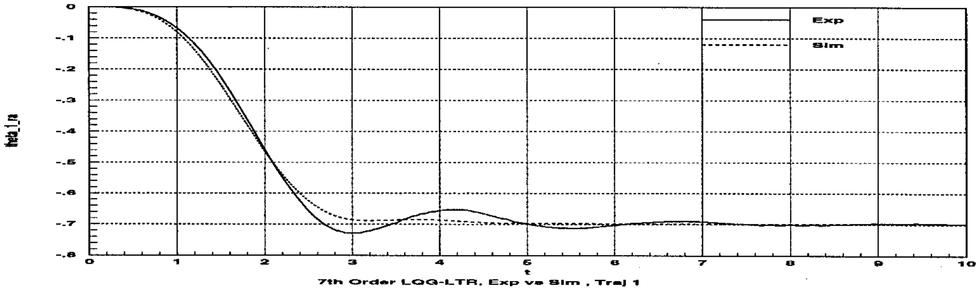


Figure C.14- 7th Order LQG/LTR Experimental vs. Sim, Trajectory 1, θ_1

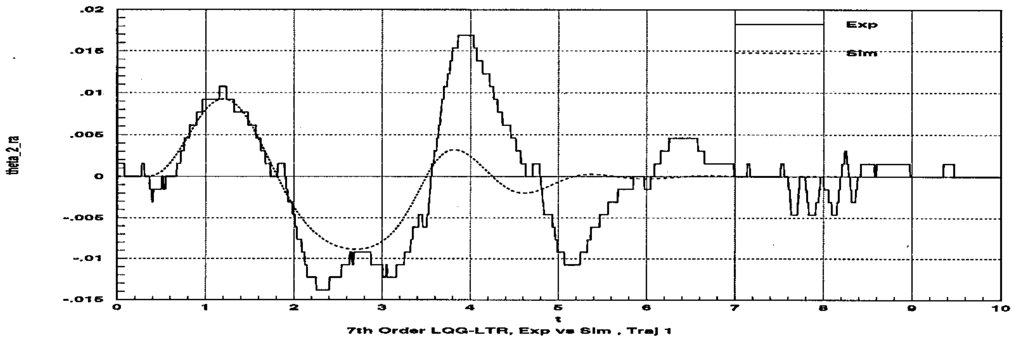


Figure C.15- 7th Order LQG/LTR Experimental vs. Sim, Trajectory 1, θ_2

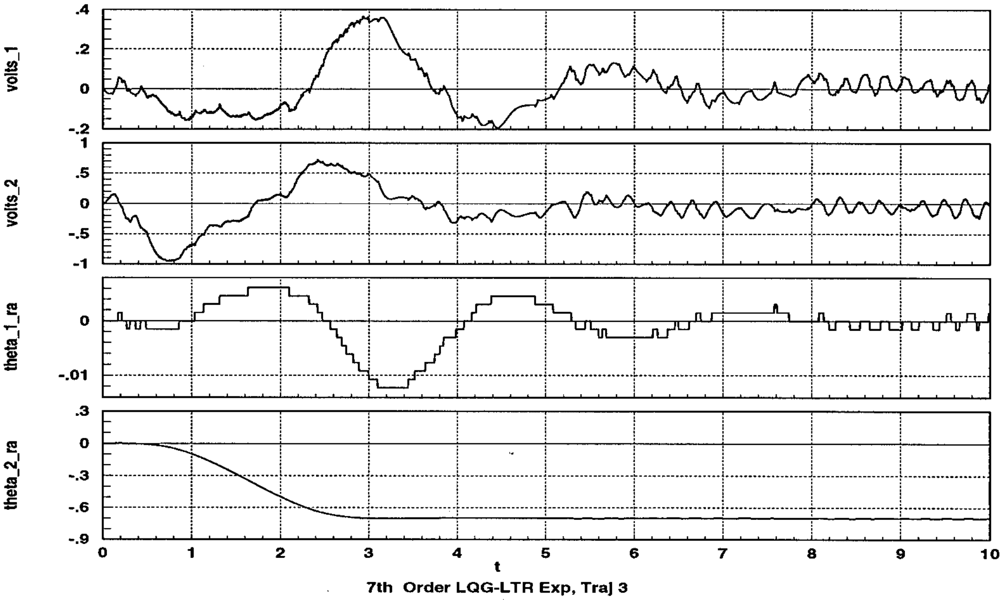


Figure C.16- 7th Order LQG/LTR Experimental Run, Trajectory 3

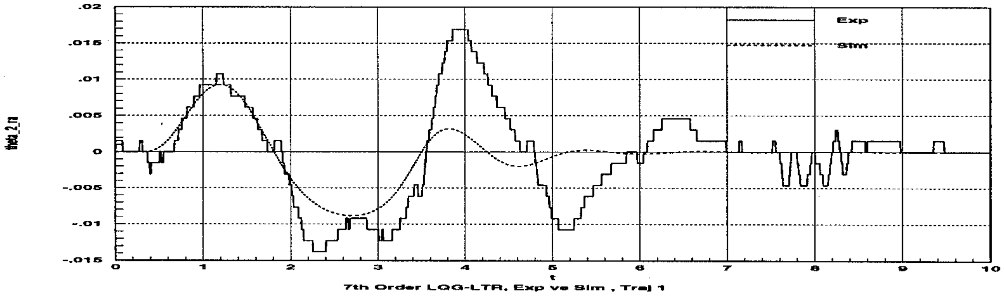


Figure C.17- 7th Order LQG/LTR Experimental vs. Simulation, Traj 3, θ_1

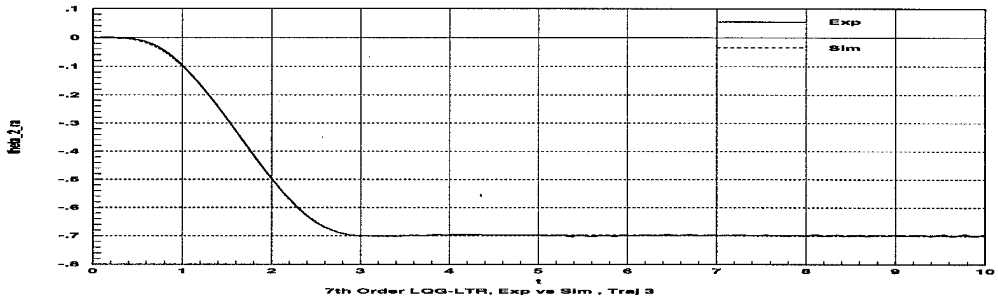


Figure C.18- 7th Order LQG/LTR Experimental vs. Simulation, Traj 3, θ_2

C.4 7th Order LTR Plots

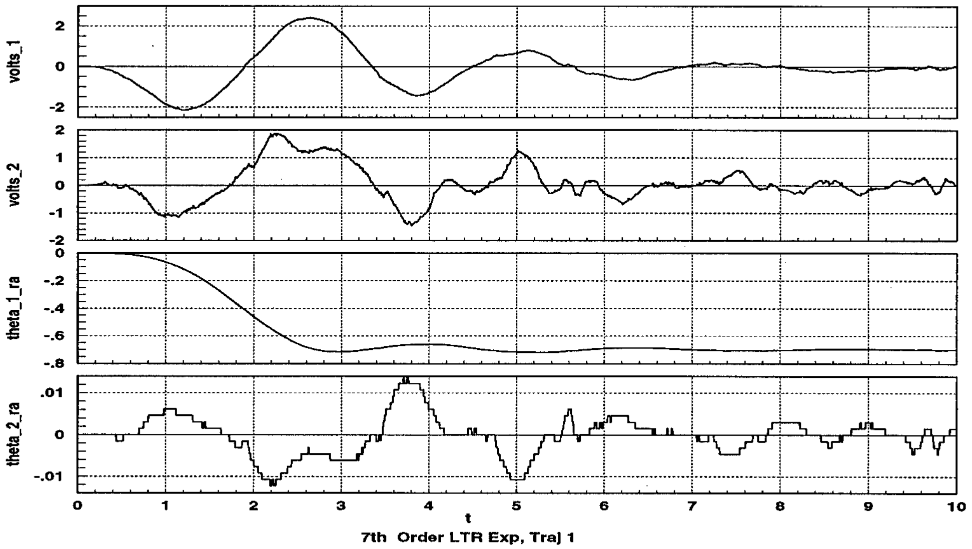


Figure C.19- 7th Order LTR Experimental Run, Trajectory 1

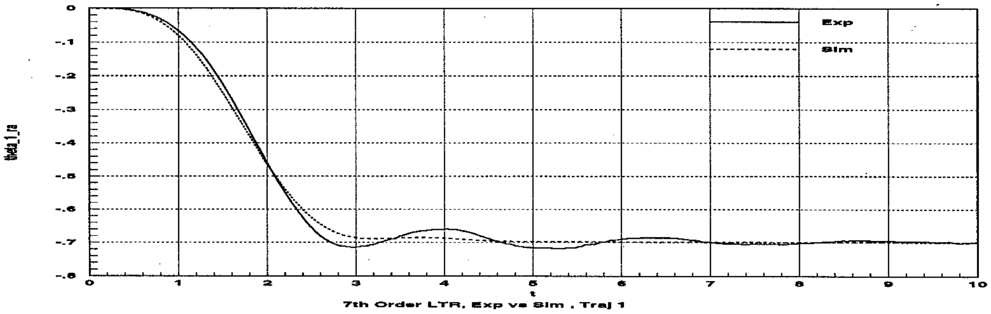


Figure C.20- 7th Order LTR Experimental vs. Simulation, Trajectory 1, θ_1

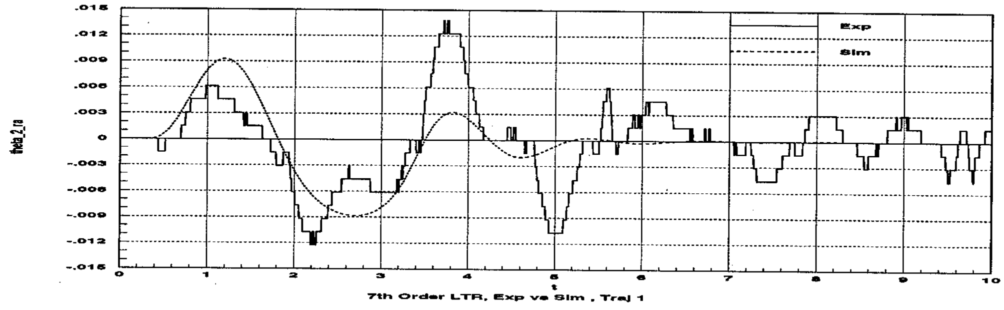


Figure C.21- 7th Order LTR Experimental vs. Simulation, Trajectory 1, θ_2

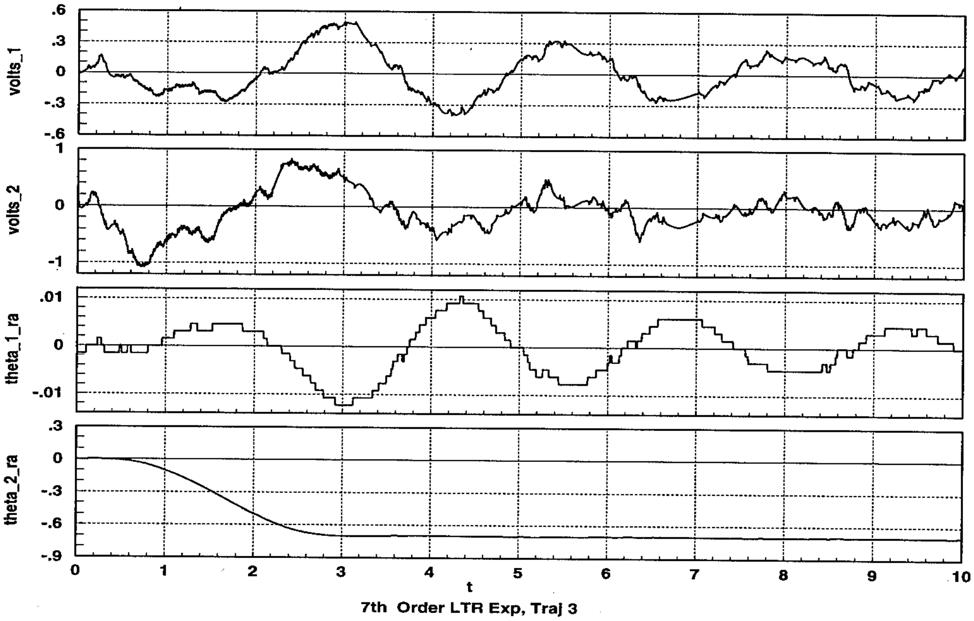


Figure C.22- 7th Order LTR Experimental Run, Trajectory 3

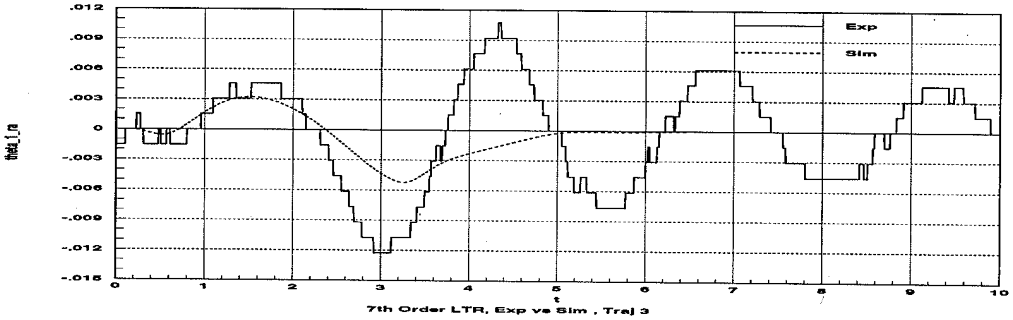


Figure C.23- 7th Order LTR Experimental vs. Simulation, Trajectory 3, θ_1

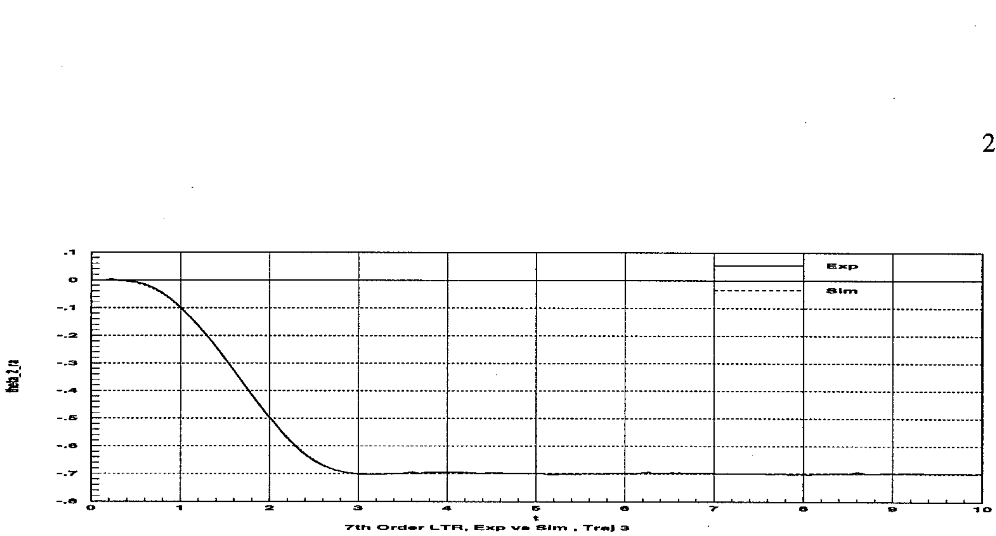


Figure C.24- 7th Order LTR Experimental vs. Simulation, Trajectory 3, θ_2

C.5 Overall Comparison Plots

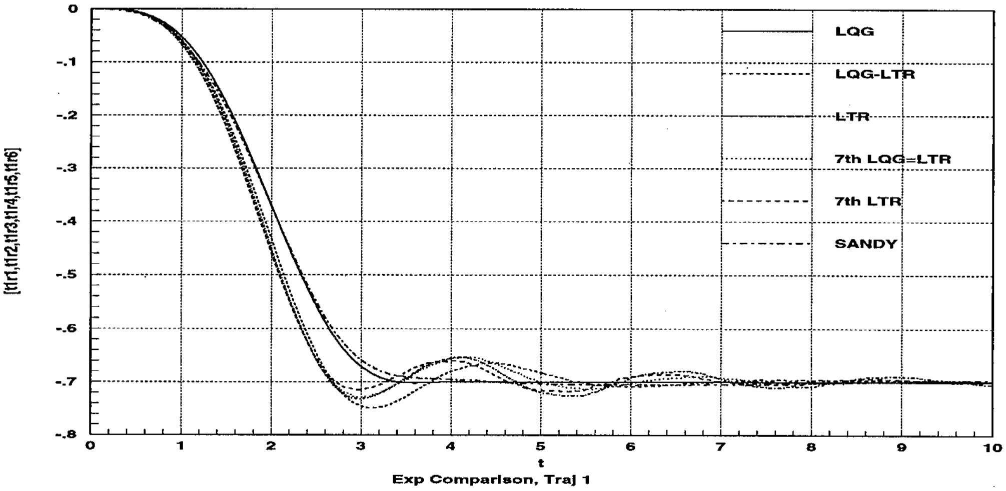


Figure C.25- Comparison Experimental Runs, Trajectory 1, θ_1

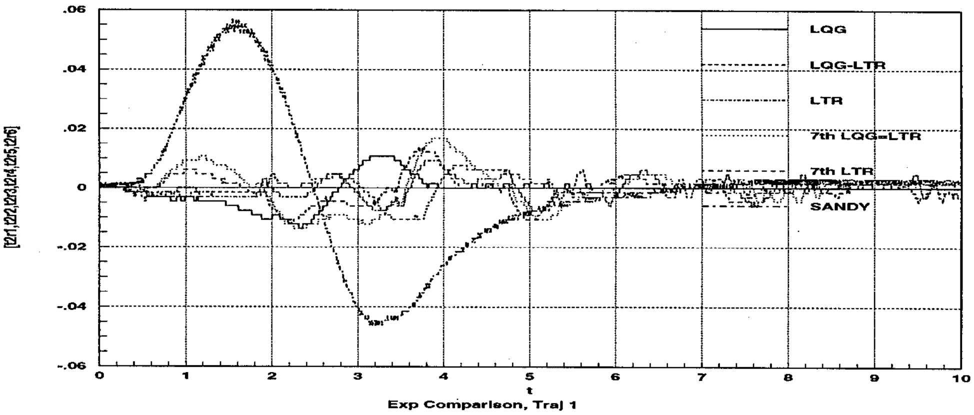


Figure C.26- Comparison Experimental Runs, Trajectory 1, θ_2

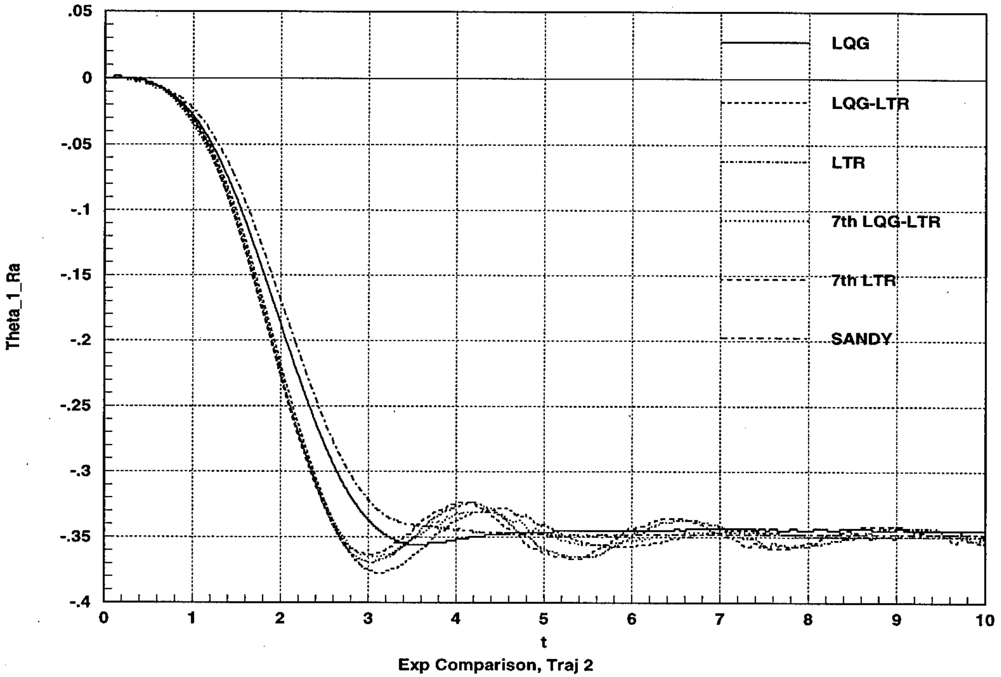


Figure C.27- Comparison Experimental Runs, Trajectory 2, θ_1

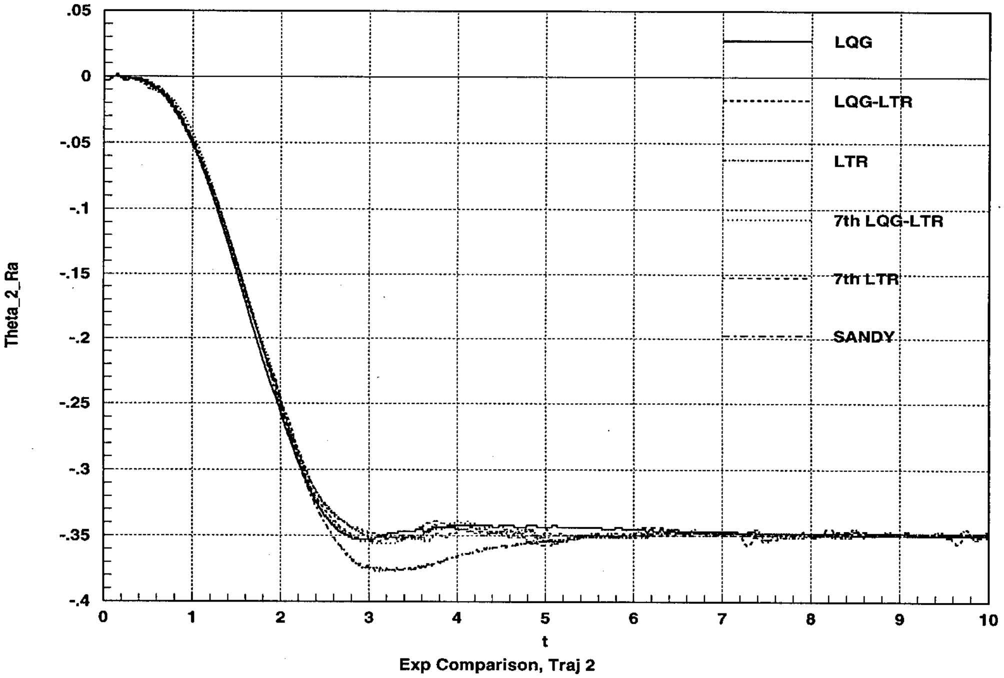


Figure C.28- Comparison Experimental Runs, Trajectory 2, θ_2

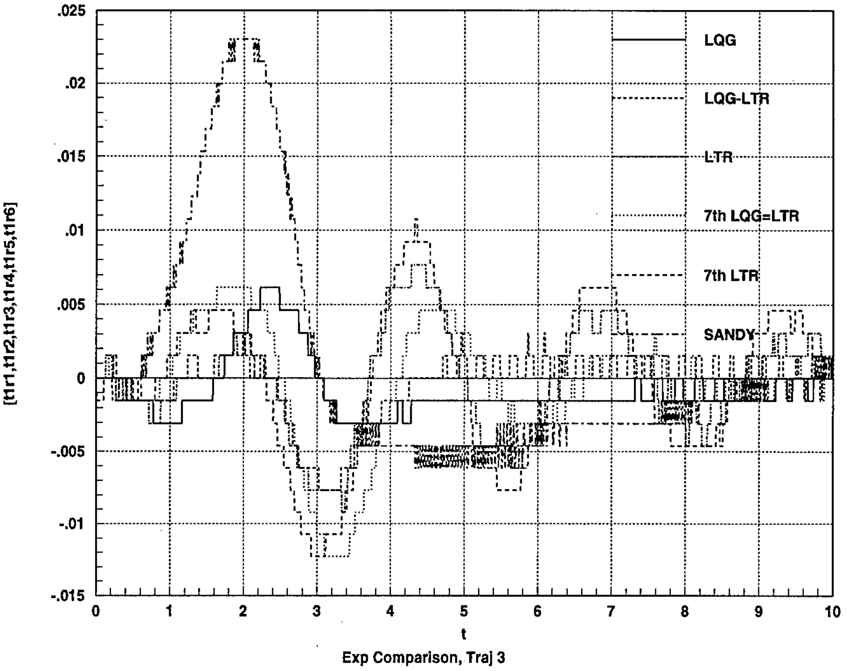


Figure C.29- Comparison Experimental Runs, Trajectory 3, θ_1

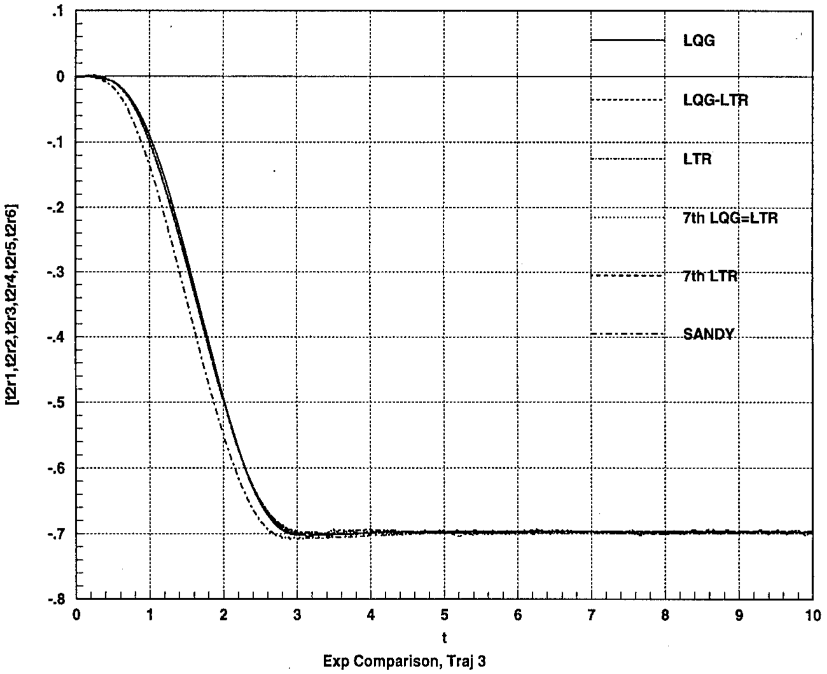
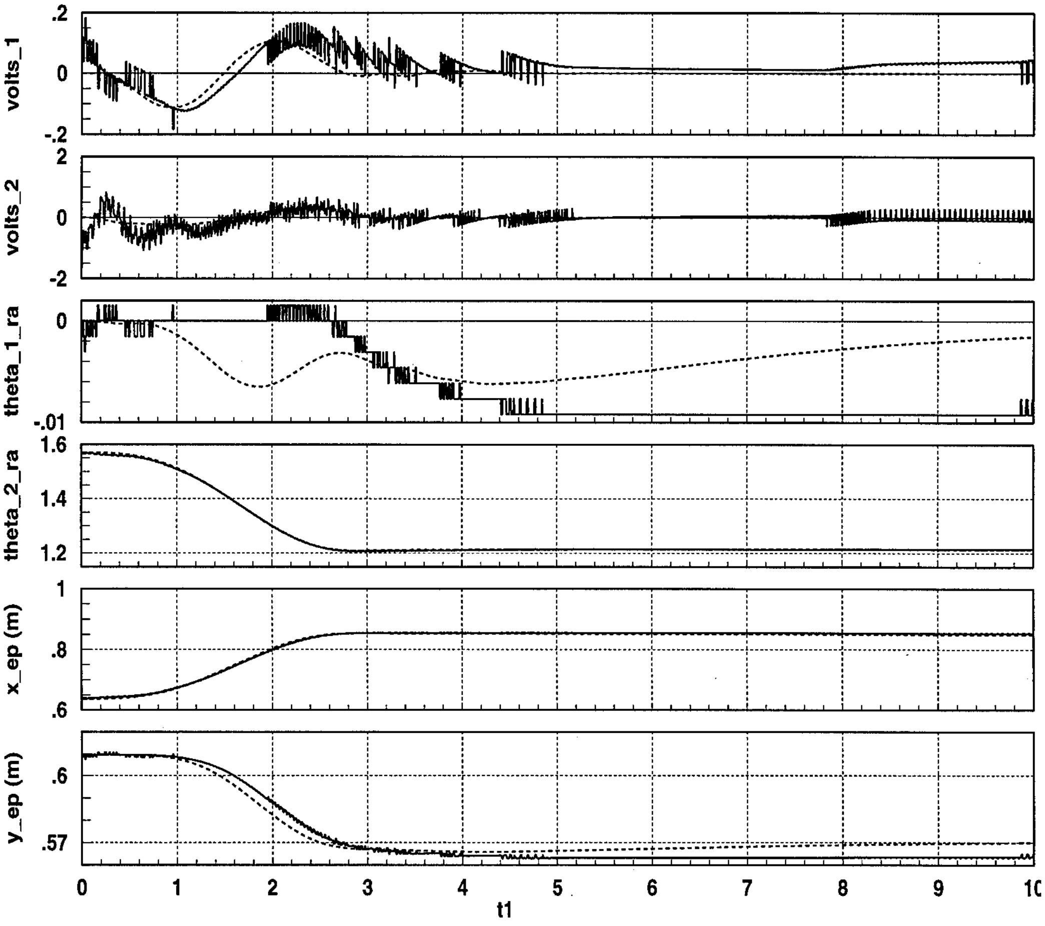


Figure C.30- Comparison Experimental Runs, Trajectory 3, θ_2

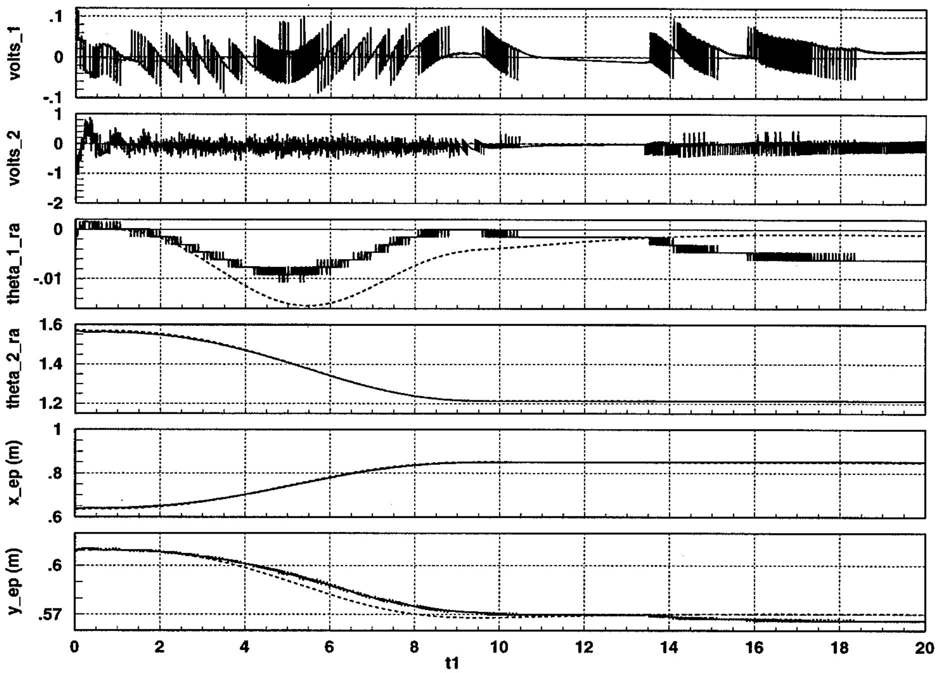
Appendix D - Additional Endpoint Tracking Plots

This appendix provides a further reference for the trajectories described in Chapter 9. Specifically, 3 and 10-second slew plots for the line and arc trajectories are provided, and 6 and 20-second slew plots for the sine trajectory are included. The plots list the x and y endpoint positions, hub angles, and voltage output for each experimental run. Also listed on each plot are the simulation results. In all the plots with volts listed, the conversion for τ_1 volts to Nm is 1/1.9208 and the conversion for τ_2 volts to Nm is 1/8.045. The choppiness of some of the angle plots is due to angle encoder quantization.



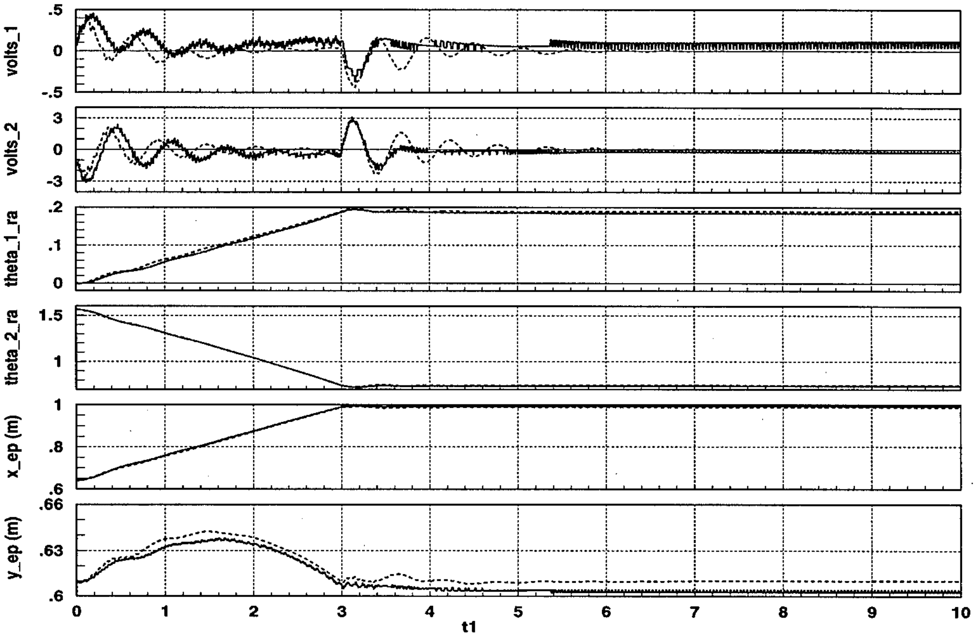
.21 m Line, Exp vs. Sim, 3 Sec Slew, Line=Exp, Dash=Sim

Figure D.1- Line Traj, 3 Sec Slew, Exp. vs. Sim.



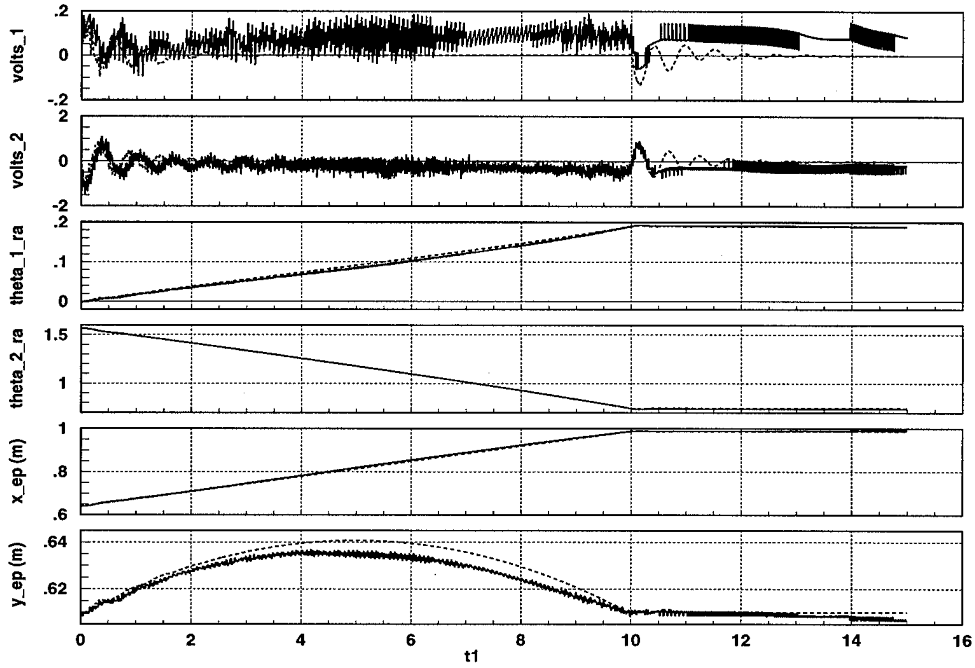
.21m Line, Exp vs. Sim, 10 Sec Slew, Line=Exp, Dash=Sim

Figure D.2- Line Traj, 10 Sec Slew, Exp. vs. Sim.



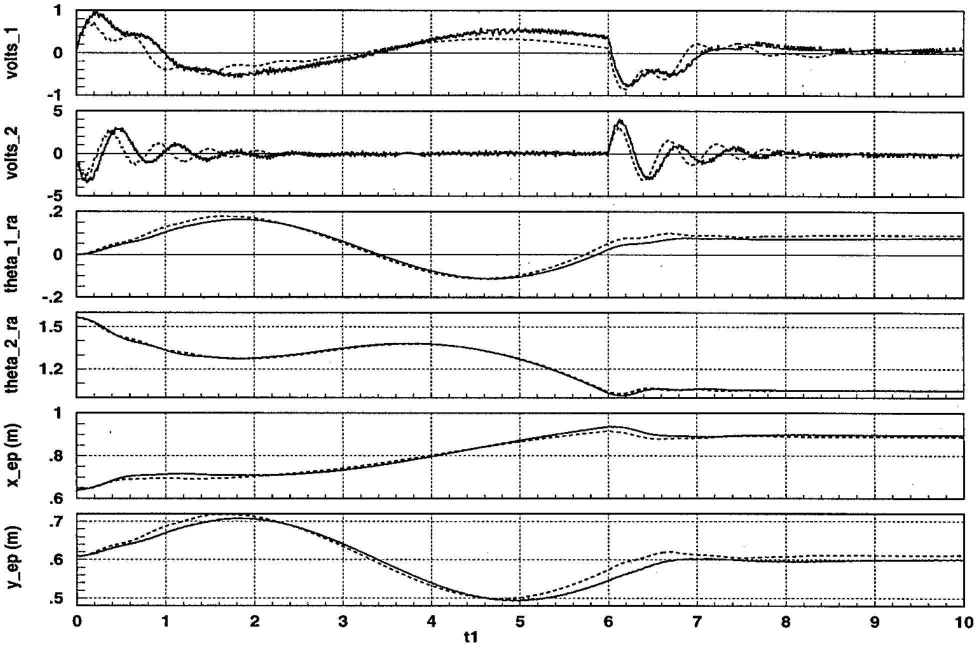
40 deg Arc, Exp vs Sim, 3 Second Slew, Solid=Exp, Dash=Sim

Figure D.3- Arc Traj, 3 Sec Slew, Exp. vs. Sim.



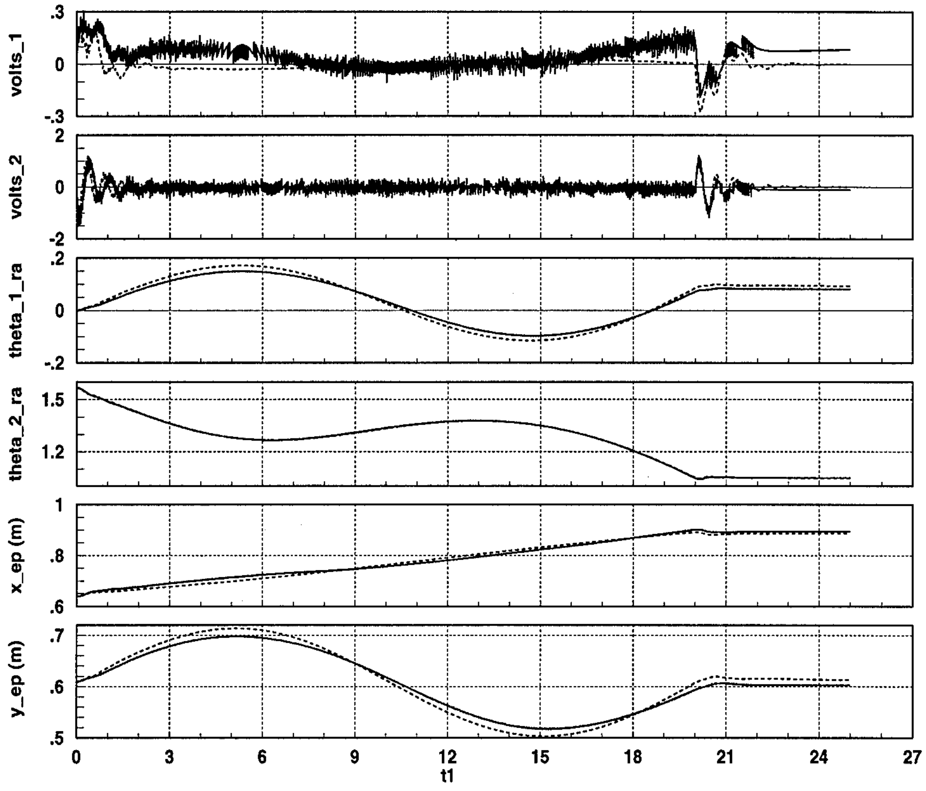
40 deg Arc, Exp vs Sim, 10 Second Slew, Solid=Exp, Dash=Sim

Figure D.4- Arc Traj, 10 Sec Slew, Exp. vs. Sim.



Sine, Amp=.25m, Length=.1m, 6 Sec Slew Exp vs Sim, Line=Exp, Dash=Sim

Figure D.5- Sine Traj, 6 Sec Slew, Exp. vs. Sim.



Sine, Amp=.25m, Length=.1m, 20 Sec Slew Exp vs Sim, Line=Exp, Dash=Sim

Figure D.6- Sine Traj, 20 Sec Slew, Exp. vs. Sim.

Appendix E- MATLAB m.files

This appendix contains the MATLAB m.files used to generate the models and controllers for this dissertation. There are three general categories for the m.files - model generation, SANDY design, and general design. The model is generated with the help of two m.files. First, the **tlapar3.m** file loads the desired parameters into the matlab workspace, and then **tlagen.m** file generates a linearized state-space model based on the loaded parameters.

The SANDY design is primarily accomplished with the help of two m.files. First, the **sandyarm.m** file generates a synthesis model compatible with SANDY. Next, the synthesis model is used in **sanarmin.m** to generate a SANDY data file.

The general design files are used in LQG design and also controller analysis. Specifically, **lqgarmdesign.m** is used to the design and reduce the LQG-class controllers. It also analyzes the controller bandwidths. The **bw.m** file analyzes the bandwidths of the PID and SANDY controllers.

E.1 Model Generation m.files

The following two m.files are used to generate the lumped-spring-mass-damper model for the two-link flexible manipulator.

E.1.1 tlapar3.m

```
% function
[Ih1,Il1,It1,mh1,ml1,mt1,ch1,oh1,l1,ct1,ot1,k1,b1,Ih2,Il2,It2,mh2,ml2,mt2,ch2,oh2
,l2,ct2,ot2,k2,b2] = plant_param_f(n1,n2)
    % Description:  Returns TLA inertial and geometric parameters.  The
    % configuration is for 2 flexible links with no mass intensifiers.  The elbow
    % motor stator is part of the tip.
    % Format: [Ih1,Il1,It1,mh1,ml1,mt1,ch1,oh1,l1,ct1,ot1,k1,
    %         Ih2,Il2,It2,mh2,ml2,mt2,ch2,oh2,l2,ct2,ot2,k2] =
    %         TLA_plant_param_f(n1,n2)
    % Output Variables
% Ihi = Inertia of ith hub about its mass center
% Ili = Inertia of ith link about its mass center
% Iti = Inertia of ith tip about its mass center
% mhi = mass of ith hub
% mli = mass of ith link
% mti = mass of ith tip
% chi = distance from ith motor shaft to ith hub center of mass
% cti = distance from ith motor shaft to ith tip center of mass
% ohi = distance from ith motor shaft to ith hub attatch point of link i
% oti = distance from ith motor shaft to ith tip attatch point of link i
% li = length of each rigid segment of link i
% ki = torsional spring stiffness between ith link segments
% bi = torsional viscous damping between ith link segments
% Revision: 7 Oct 94, Dave Bossert
% by Steve Evers
% Variables
n1=10
n2=10
th1_nom=0*pi/180
th2_nom=90*pi/180
% shoulder hub parameters
Ih1 = 4.1238e-03; %kg*m^2
mh1 = 1.1618; %kg
ch1 = 5.5959e-05; %m
oh1 = 5.8420e-02; %m
% link 1 parameters
% long flexible link, no mass intensifiers
ml1 = (169.5/1000)*((21-1-1.5)/21); %kg (measured)
ml1 = ml1/n1; %kg
l1 = ((21-1-1.5)/12)*0.3048; %m (measured)
```

```
l1 = l1/n1; %m
Il1 = (ml1*l1^2)/12; %kg*m^2
% link 2 parameters
% long flexible link, no mass intensifiers
ml2 = (169.5/1000)*((20.5-1-1)/21); %kg (measured)
ml2 = ml2/n2; %kg
l2 = ((20.5-1-1)/12)*0.3048; %m (measured)
l2 = l2/n2; %m
Il2 = (ml2*l2^2)/12; %kg*m^2
% tip includes standing bracket, stator housing and pad
% roughly experimentally measured
% swing test with cm assumed at joint
It1 = 0.0058;
mt1 = 2.0755;
ct1 = 4.3807e-04;
ot1 = 0.0063; %m
% hub includes hanging bracket, shaft coupling and extender
Ih2 = 5.1099e-04;
mh2 =  0.5745;
ch2 = 0.0236;
oh2 = 4.5*.0254;
% tip 2 parameters
% tip is composed of bracket, pad, and link grip length
% tip location is defined as centerline of bracket base bolts
% same as first SLA tip
mt2 = .35; %2.9842e-01; %kg
ct2 = 4.3807e-04; %m
ot2 = 6.3500e-03; %m
It2 = .003; %1.1045e-03; %kg*m^2
% beam lateral stiffness properties
% Experimentally tuned
EIl1 = 5.5;
EIl2 = EIl1;
k1 = 3*EIl1*(n1*(n1-1)*(2*n1-1)/6)/(n1^2*n1*l1);
k2 = 3*EIl2*(n2*(n2-1)*(2*n2-1)/6)/(n2^2*n2*l1);
% beam structural damping
% experimentally tuned to 40th order plant model 90 deg joint
b = 0.01;
b1 = 60*b/n1;
b2 = (60*b/n2)/2;
% end
```

E.1.2 tlagen.m

```
% function [Abar,Bbar,C,D,M,k1,k2,b1,b2] = ...
% plant_gen_f(n1,n2,th1_nom,th2_nom)
% Format:  [A,B,C,D] = TLA_plant_gen_f(n1,n2,th1_nom,th2_nom)
%
% Description:  Synthesizes a lumped spring/mass/damper (LSMD) state space
%               model for the UW flex-flex TLA in the following form.  Check
%               file for configuration case:
%
%                 d/dt{x} = A{x} + B{u}
%                 {del_y} = C{x} + D{u}
%
%               Note: both {x} and {del_y} are referenced to the LOCAL
%               coordinate system determined by the state vector:
%                 x = {0 ... 0 th1_nom 0 ... 0 th2_nom 0 ... 0}'
%
%  x = {dth1 dq11 ... dq1(n1-1) dth2 dq21 ... dq2(n2-1) th1 ... q2(n2-1)}'
%  {del_y} = {x del_xep del_yep}'
```

```
    %  {u} = {T1 T2}'
    %  T1 = shoulder torque
    %  T2 = elbow torque
    %  th1 = shoulder joint perturbed base angle
    %  th2 = elbow joint perturbed base angle
    %  del_xep = x end point perturbed cartesian position
    %  del_yep = y end point perturbed cartesian position
    %
    %  ALL UNITS IN METRIC (kg-m-s)
    %
    % Input Variables:  n1 = number of flexible elements for first link
    %                   n2 = number of flexible elements for second link
    %                   th1_nom = nominal shoulder joint angle to linearize about
    %                   th2_nom = nominal elbow joint angle to linearize about
    %
    % Output Variables:  [A,B,C,D] = state space realization in locally prescribed
    %                                coordinate system.
    % Revision: 7 Oct 94 by Dave Bossert
    % by Steve Evers
    % load inertia/spring/mass parameters
    %
[Ih1,Il1,It1,mh1,ml1,mt1,ch1,oh1,l1,ct1,ot1,k1,b1,Ih2,Il2,It2,mh2,ml2,mt2,ch2,oh2
,l2,ct2,ot2,k2,b2] = plant_param_f(n1,n2);
    % mass matrices
    for i = 2:n1-1,
      M_inertias11(i,i) = Il1;
    end
    M_inertias11(1,1) = Ih1+Il1;
    M_inertias11(n1,n1) = Il1+It1;
    for i = 2:n2-1,
 M_inertias22(i,i) = Il2;
end
M_inertias22(1,1) = Ih2+Il2;
M_inertias22(n2,n2) = Il2+It2;
M_inertias = [M_inertias11 zeros(n1,n2);zeros(n2,n1) M_inertias22];
for i = 2:n1-1,
 M_hubs_diag11(i,i) = mh2*l1^2;
end
M_hubs_diag11(1,1) = mh1*ch1^2+mh2*(oh1+l1)^2;
M_hubs_diag11(n1,n1) = mh2*(l1+ot1)^2;
M_hubs_diag22 = zeros(n2,n2);
M_hubs_diag22(1,1) = mh2*ch2^2;
M_hubs_diag = [M_hubs_diag11 zeros(n1,n2);zeros(n2,n1) M_hubs_diag22];
for i = 2:n1,
 M_link1_diag11(i,i) = (1/4)*ml1*l1^2+(n1-i)*ml1*l1^2;
end
% Stanford element here is incorrect
%M_link1_diag11(1,1) = ml1*(oh1+(l1/2))^2+n1*ml1*(oh1+l1)^2;
% My element
M_link1_diag11(1,1) = ml1*(oh1+(l1/2))^2+(n1-1)*ml1*(oh1+l1)^2;
M_link1_diag = [M_link1_diag11 zeros(n1,n2);zeros(n2,n1+n2)];
for i = 2:n1-1,
 M_link2_diag11(i,i) = n2*ml2*l1^2;
end
M_link2_diag11(1,1) = n2*ml2*(oh1+l1)^2;
M_link2_diag11(n1,n1) = n2*ml2*(l1+ot1)^2;
for i = 2:n2,
 M_link2_diag22(i,i) = (1/4)*ml2*l2^2+(n2-i)*ml2*l2^2;
end
% Stanford element here is incorrect
```

```
%M_link2_diag22(1,1) = ml2*(oh2+(l2/2))^2+n2*ml2*(oh2+l2)^2;
% My element
M_link2_diag22(1,1) = ml2*(oh2+(l2/2))^2+(n2-1)*ml2*(oh2+l2)^2;
M_link2_diag = [M_link2_diag11 zeros(n1,n2);zeros(n2,n1) M_link2_diag22];
M_tip1_diag = zeros((n1+n2),(n1+n2));
for i = 2:n1-1,
 M_tip1_diag(i,i) = mt1*l1^2;
end
M_tip1_diag(1,1) = mt1*(oh1+l1)^2;
M_tip1_diag(n1,n1) = mt1*(l1+ot1+ct1)^2;
for i = 2:n1-1,
 M_tip2_diag11(i,i) = mt2*l1^2;
end
M_tip2_diag11(1,1) = mt2*(oh1+l1)^2;
M_tip2_diag11(n1,n1) = mt2*(l1+ot1)^2;
for i = 2:n2-1,
 M_tip2_diag22(i,i) = mt2*l2^2;
end
M_tip2_diag22(1,1) = mt2*(oh2+l2)^2;
M_tip2_diag22(n2,n2) = mt2*(l2+ot2+ct2)^2;
M_tip2_diag = [M_tip2_diag11 zeros(n1,n2);zeros(n2,n1) M_tip2_diag22];
M_hubs_off_diag11 = zeros(n1,n1);
for i = 2:n1-2,
 for j = i+1:n1-1,
  M_hubs_off_diag11(i,j) = mh2*l1^2;
 end
end
for i = 2:n1-1,
 M_hubs_off_diag11(1,i) = mh2*(oh1+l1)*l1;
 M_hubs_off_diag11(i,n1) = mh2*l1*(l1+ot1);
```

```
end
M_hubs_off_diag11(1,n1) = mh2*(oh1+l1)*(l1+ot1);
M_hubs_off_diag11 = M_hubs_off_diag11+M_hubs_off_diag11’;
M_hubs_off_diag12 = zeros(n1,n2);
for i = 2:n1-1,
 M_hubs_off_diag12(i,1) = mh2*l1*ch2;
end
M_hubs_off_diag12(1,1) = mh2*(oh1+l1)*ch2;
M_hubs_off_diag12(n1,1) = mh2*(l1+ot1)*ch2;
M_hubs_off_diag = [M_hubs_off_diag11 M_hubs_off_diag12*cos(th2_nom);
...
          (M_hubs_off_diag12’)*cos(th2_nom) zeros(n2,n2)];
M_link1_off_diag11 = zeros(n1,n1);
for i = 2:n1-1,
 for j = i+1:n1,
  M_link1_off_diag11(i,j) = (1/2)*ml1*l1^2+(n1-j)*ml1*l1^2;
 end
end
for i = 2:n1,
 M_link1_off_diag11(1,i) = (1/2)*ml1*(oh1+l1)*l1+(n1-i)*ml1*(oh1+l1)*l1;
end
M_link1_off_diag11 = M_link1_off_diag11+M_link1_off_diag11’;
M_link1_off_diag = [M_link1_off_diag11 zeros(n1,n2); zeros(n2,n1+n2)];
M_link2_off_diag11 = zeros(n1,n1);
for i = 2:n1-2,
 for j = i+1:n1-1,
  M_link2_off_diag11(i,j) = n2*ml2*l1^2;
 end
end
for i = 2:n1-1,
  M_link2_off_diag11(1,i) = n2*ml2*(oh1+l1)*l1;
  M_link2_off_diag11(i,n1) = n2*ml2*l1*(l1+ot1);
end
M_link2_off_diag11(1,n1) = n2*ml2*(oh1+l1)*(l1+ot1);
M_link2_off_diag11 = M_link2_off_diag11+M_link2_off_diag11';
for i = 2:n1-1,
  for j = 2:n2,
    M_link2_off_diag12(i,j) = (1/2)*ml2*l1*l2+(n2-j)*ml2*l1*l2;
  end
end
for i = 2:n2,
  M_link2_off_diag12(1,i) = (1/2)*ml2*(oh1+l1)*l2+(n2-i)*ml2*(oh1+l1)*l2;
end
for i = 2:n1-1,
  M_link2_off_diag12(i,1) = ml2*l1*(oh2+(l2/2))+(n2-1)*ml2*l1*(oh2+l2);
end
for i = 2:n2,
  M_link2_off_diag12(n1,i) = (1/2)*ml2*(l1+ot1)*l2+(n2-i)*ml2*(l1+ot1)*l2;
end
M_link2_off_diag12(1,1) = ml2*(oh1+l1)*(oh2+(l2/2))+(n2-1)*ml2*(oh1+l1)*(oh2+l2);
M_link2_off_diag12(n1,1) = ml2*(l1+ot1)*(oh2+(l2/2))+(n2-1)*ml2*(l1+ot1)*(oh2+l2);
M_link2_off_diag22 = zeros(n2,n2);
for i = 2:n2-1,
  for j = i+1:n2,
    M_link2_off_diag22(i,j) = (1/2)*ml2*l2^2+(n2-j)*ml2*l2^2;
  end
end
for i = 2:n2,
 M_link2_off_diag22(1,i) = (1/2)*ml2*(oh2+l2)*l2+(n2-i)*ml2*(oh2+l2)*l2;
end
M_link2_off_diag22 = M_link2_off_diag22+M_link2_off_diag22’;
M_link2_off_diag = [M_link2_off_diag11 M_link2_off_diag12*cos(th2_nom);
...
          (M_link2_off_diag12’)*cos(th2_nom) M_link2_off_diag22];
M_tip1_off_diag11 = zeros(n1,n1);
for i = 2:n1-2,
 for j = i+1:n1-1,
  M_tip1_off_diag11(i,j) = mt1*l1^2;
 end
end
for i = 2:n1-1,
 M_tip1_off_diag11(1,i) = mt1*(oh1+l1)*l1;
 M_tip1_off_diag11(i,n1) = mt1*l1*(l1+ot1+ct1);
end
M_tip1_off_diag11(1,n1) = mt1*(oh1+l1)*(l1+ot1+ct1);
M_tip1_off_diag11 = M_tip1_off_diag11+M_tip1_off_diag11’;
M_tip1_off_diag = [M_tip1_off_diag11 zeros(n1,n2);zeros(n2,n1+n2)];
M_tip2_off_diag11 = zeros(n1,n1);
for i = 2:n1-2,
 for j = i+1:n1-1,
  M_tip2_off_diag11(i,j) = mt2*l1^2;
 end
end
for i = 2:n1-1,
 M_tip2_off_diag11(1,i) = mt2*(oh1+l1)*l1;
 M_tip2_off_diag11(i,n1) = mt2*l1*(l1+ot1);
end
M_tip2_off_diag11(1,n1) = mt2*(oh1+l1)*(l1+ot1);
```

```
M_tip2_off_diag11 = M_tip2_off_diag11+M_tip2_off_diag11’;
M_tip2_off_diag12 = zeros(n1,n2);
for i = 2:n1-1,
 for j = 2:n2-1,
   M_tip2_off_diag12(i,j) = mt2*l1*l2;
 end
  M_tip2_off_diag12(i,1) = mt2*l1*(oh2+l2);
  M_tip2_off_diag12(i,n2) = mt2*l1*(l2+ot2+ct2);
end
for i = 2:n2-1,
M_tip2_off_diag12(1,i) = mt2*(oh1+l1)*l2;
M_tip2_off_diag12(n1,i) = mt2*(l1+ot1)*l2;
end
M_tip2_off_diag12(1,1) = mt2*(oh1+l1)*(oh2+l2);
M_tip2_off_diag12(1,n2) = mt2*(oh1+l1)*(l2+ot2+ct2);
M_tip2_off_diag12(n1,1) = mt2*(l1+ot1)*(oh2+l2);
M_tip2_off_diag12(n1,n2) = mt2*(l1+ot1)*(l2+ot2+ct2);
M_tip2_off_diag22 = zeros(n2,n2);
for i = 2:n2-2,
 for j = i+1:n2-1,
   M_tip2_off_diag22(i,j) = mt2*l2^2;
 end
end
for i = 2:n2-1,
 M_tip2_off_diag22(1,i) = mt2*(oh2+l2)*l2;
 M_tip2_off_diag22(i,n2) = mt2*l2*(l2+ot2+ct2);
end
M_tip2_off_diag22(1,n2) = mt2*(oh2+l2)*(l2+ot2+ct2);
M_tip2_off_diag22 = M_tip2_off_diag22+M_tip2_off_diag22’;
M_tip2_off_diag = [M_tip2_off_diag11 (M_tip2_off_diag12)*cos(th2_nom); ...
```

```
          (M_tip2_off_diag12')*cos(th2_nom) M_tip2_off_diag22];
  M                                                                  =
M_inertias+M_hubs_diag+M_link1_diag+M_link2_diag+M_tip1_diag+M_tip2_di
ag+M_hubs_off_diag+M_link1_off_diag+M_link2_off_diag+M_tip1_off_diag+M
_tip2_off_diag;
  % state transformation of mass matrix u to x
  for i = 1:(n1+n2),
   for j = 1:(n1+n2),
    Mbar(i,j) = sum(M(i,j:(n1+n2)));
   end
  end
  % stiffness matrix
  for i = 1:(n1-1),
   K1(i,i+1) = -k1;
   K1(i+1,i+1) = k1;
  end
  for i = 1:(n2-1),
   K2(i,i+1) = -k2;
   K2(i+1,i+1) = k2;
  end
  K = [K1 zeros(n1,n2);zeros(n2,n1) K2];
  % damping matrix
  for i = 1:(n1-1),
   D1(i,i+1) = -b1;
   D1(i+1,i+1) = b1;
  end
  for i = 1:(n2-1),
   D2(i,i+1) = -b2;
   D2(i+1,i+1) = b2;
  end
```

```
    D = [D1 zeros(n1,n2);zeros(n2,n1) D2];

    % input torque matrix
    T = zeros(n1+n2,2);
    T(1,1) = 1;
    T(n1,2) = -1;
    T(n1+1,2) = 1;
    % state space realization
    % Mbar{d/dt(x)} = A{x} + T{u}
    % {y} = C{x} + D{u}
    Mbar = [Mbar zeros(n1+n2);
          zeros(n1+n2) eye(n1+n2)];
    A = [-1*D -1*K;
       eye(n1+n2) zeros(n1+n2)];
    T = [T;zeros(n1+n2,2)];
    Abar = Mbar\A;
    Bbar = Mbar\T;
    C = [eye(2*(n1+n2));
       zeros(2,2*(n1+n2))];
    % Stanford model had error in the term relating to the 1st element of each link
    C(2*(n1+n2)+1,n1+n2+1) = -(oh1+l1)*sin(th1_nom)-(n1-1)*l1*sin(th1_nom)-
ot1*sin(th1_nom)-(oh2+l2)*sin(th1_nom+th2_nom)-(n2-
1)*l2*sin(th1_nom+th2_nom)-ot2*sin(th1_nom+th2_nom);
    cc              =              -(n1-1)*l1*sin(th1_nom)-ot1*sin(th1_nom)-
(oh2+l2)*sin(th1_nom+th2_nom)-(n2-1)*l2*sin(th1_nom+th2_nom)-
ot2*sin(th1_nom+th2_nom);
    for i = 1:(n1-1),
     C(2*(n1+n2)+1,n1+n2+i+1) = cc-(1-i)*l1*sin(th1_nom);
    end
    C(2*(n1+n2)+1,n1+n2+n1+1)     =     -(oh2+l2)*sin(th1_nom+th2_nom)-(n2-
1)*l2*sin(th1_nom+th2_nom)-ot2*sin(th1_nom+th2_nom);
    cc = -(n2-1)*l2*sin(th1_nom+th2_nom)-ot2*sin(th1_nom+th2_nom);
    for i = 1:(n2-1),
     C(2*(n1+n2)+1,n1+n2+i+n1+1) = cc-(1-i)*l2*sin(th1_nom+th2_nom);
    end
    C(2*(n1+n2)+2,n1+n2+1) = (oh1+l1)*cos(th1_nom)+(n1-
1)*l1*cos(th1_nom)+ot1*cos(th1_nom)+(oh2+l2)*cos(th1_nom+th2_nom)+(n2-
1)*l2*cos(th1_nom+th2_nom)+ot2*cos(th1_nom+th2_nom);
    cc = (n1-
1)*l1*cos(th1_nom)+ot1*cos(th1_nom)+(oh2+l2)*cos(th1_nom+th2_nom)+(n2-
1)*l2*cos(th1_nom+th2_nom)+ot2*cos(th1_nom+th2_nom);
    for i = 1:(n1-1),
     C(2*(n1+n2)+2,n1+n2+i+1) = cc+(1-i)*l1*cos(th1_nom);
    end
    C(2*(n1+n2)+2,n1+n2+n1+1) = (oh2+l2)*cos(th1_nom+th2_nom)+(n2-
1)*l2*cos(th1_nom+th2_nom)+ot2*cos(th1_nom+th2_nom);
    cc = (n2-1)*l2*cos(th1_nom+th2_nom)+ot2*cos(th1_nom+th2_nom);
    for i = 1:(n2-1),
     C(2*(n1+n2)+2,n1+n2+i+n1+1) = cc+(1-i)*l2*cos(th1_nom+th2_nom);
    end
    D = zeros(2*(n1+n2)+2,2);
    % end
```

E.2 SANDY Design m.files

The following two m.files are used in the SANDY controller design process.

E.2.1 sandyarm.m

```
    % sandyarm.m
    % load plant model, and form sandy synthesis model
    %
    % Full order, 20 DOF
```

```
% load 40ordarmev
% Reduced order, 20 DOF Evers
% load 18ordarmev
% reduced order, 20 DOF fast
% load 18ordarmhs
% Reducer order, 20 DOF
% Written by Dave Bossert
% Last Revsion: 7 Oct 94
load 30ordarm
% start blocks for the synthesis model
% Block 1 - Plant  (much slower than actuators, therefore no actuator dynamics
included)
a1=Abar;
b1=Bbar;
c1=C;
d1=D;
% Block 2 - Theta1 command input
n2=1;
d2=1;
% Block 3 - Theta2 command input
n3=1;
d3=1;
% Block 4 - Theta1 error output
n4=1;
d4=1;
% Block 5 - Theta2 error output
n5=1;
d5=1;
% Build the synthesis model
nblocks=5;
```

```
blkbuild;
% inputs - delta_t1, delta_t2, t1_cmd, t2_cmd
inputs=[1,2,3,4];
% 40 DOF
% outputs - theta1dot, theta2dot, error_theta1, error_theta2
% outputs=[1,21,85,86];
%Q=[5,41,-83;
%   6,61,-84];
% 20 DOF with thetadots
 outputs=[1,11,45,46];
 Q=[5,-21,43;
   6,-31,44];
% 20 DOF without thetadots
% outputs=[45,46];
% Q=[5,21,-43;
%   6,31,-44];
[Asynth,Bsynth,Csynth,Dsynth]=connect(a,b,c,d,Q,inputs,outputs);
save sandysynth Asynth Bsynth Csynth Dsynth
```

E.2.2 sanamrin.m

```
%sanarmin.m
% loads SANDY synthesis model and generates a SANDY data file for use
% with the SANDY design software.
format short e;
%Load synthesis models
load sandysynth
% F State Matrices
F1=Asynth;
F2=F1;
% G Control Input Matrices
```

```
G1=Bsynth(:,1:2);
G2=G1;
% Gamma Disturbance Input Matrices
Gam1=Bsynth(:,3:4);
Gam2=Gam1;
% Wspec Disturbance Inputs
Wspec1=[0 0 1 1 1 0;0 0 0 0 0 1];
Wspec2=[0 0 0 0 0 1;0 0 1 1 1 0];
% Hc Criterion State Matrix -[Theta1_dot, Theta2_dot, Theta1_cmd,Theta2_cmd]
Hc1=Csynth;
Hc2=Hc1;
% Dcu Criterion Control Input Matrix
Dcu1=zeros(4,2); %for theta dots
% Dcu1=zeros(2,2);
Dcu2=Dcu1;
% Dcw Criterion Disturbance Input Matrix
Dcw1=[0,0;0,0;1,0;0,1]; %for theta dots
% Dcw1=[1,0;0,1];
Dcw2=Dcw1;
% Hs Sensor State Matrix  -ys=[theta1_dot, theta2_dot, error_theta_1, error_theta_2]
%Hs1=Csynth;
% Hs Sensor State Matrix  -ys=[ error_theta_1, error_theta_2]
Hs1=Csynth(3:4,:);
Hs2=Hs1;
% Dsu Sensor Control Input Matrix
%Dsu1=zeros(4,2);  %for theta dots
Dsu1=zeros(2,2);
Dsu2=Dsu1;
```

```
% Dsw Sensor Disturbance Input Matrix
%Dsw1=[0,0;0,0;1,0;0,1]; %for theta dots
Dsw1=[1,0;0,1];
Dsw2=Dsw1;
% Weighting Matrices
Q1=diag([0,0,1,0]);% for theta dots
Q2=diag([0,0,0,1]);
% Weighting Matrices
%Q1=diag([1,0]);
%Q2=diag([0,1]);
R1=diag([0,0]);
R2=diag([0,0]);
%Set up the problem for SANDY
sandygl
Nfcmax=10;
Tf=[25;25];
Tfctor=[2;2];
Npm=2;
Wp=[1;1];
F=[F1;F2];
G=[G1;G2];
%Disturbance inputs w(t)=[Theta1_cmd,Theta2_cmd]
Gam=[Gam1;Gam2];
%Measurement vector ys=[theta1_dot, theta2_dot, error_theta_1, error_theta_2]
Hs=[Hs1;Hs2];
Dsu=[Dsu1;Dsu2];
Dsw=[Dsw1;Dsw2];
%Criterion vector yc=[theta1_dot, theta2_dot, error_theta_1, error_theta_2]
%Criterion vector yc=[error_theta_1, error_theta_2] with no theta dots
Hc=[Hc1;Hc2];
```

```
Dcu=[Dcu1;Dcu2];
Dcw=[Dcw1;Dcw2];
Wspec=[Wspec1;Wspec2];
Alpha=[0;0];
Sigmamax=[-.02;1];
Zetamin=[0;0];
%Weighting matrices Q and R
Q=[Q1;Q2];
R=[R1;R2];
nDircu=[0;0];
Dircu=[];
nDircy=[0;0];
Dircy=[];
%Initial guess (arbitrary) of the LQG controller (with theta dots)
A=[0,0;0,0];
B=[1,0;0,1];
C=[
   -3.881 17.57;
   -2.235 10
 ];
 D=[
  6.614 -10.79
   1.052 2.647
 ];
%Initial guess (arbitrary) of the LQG controller (no theta dots)
%A=[0,0;0,0];
%B=[1,0;0,1];
%C=[1.698, 2.348; -0.1700, -.235];
%D=[];
% Design Parameters
```

```
nAid=0;
Aid=[];
nBid=0;
Bid=[];
nCid=4;
Cid=[
1,1,1,-100,100
1,2,1,-100,100
2,1,1,-100,100
2,2,1,-100,100];
%nDid=0;
%Did=[];
% with theta dot
%nDid=8;
%Did=[1,1,1,-100,100;
%    1,2,1,-100,100;
%    1,3,1,-100,100;
%    1,4,1,-100,100;
%    2,1,1,-100,100;
%    2,2,1,-100,100
%    2,3,1,-100,100;
%    2,4,1,-100,100];
% without theta dot in sensors
nDid=4;
Did=[1,1,1,-100,100;
    1,2,1,-100,100;
    2,1,1,-100,100;
    2,2,1,-100,100];
Nclin=0;
nLincoef=[];
```

```
Lincoef=[];
Linbnds=[];
Nudnlc=0;
sandysv('arminput')
```

E.3 General Design

This section presents several design m.files used for this dissertation. First, lqgarmdes.m presents LQG-class controller design and evaluation. Also, the bw.m file shows the bandwidth generation for the SANDY and PID controllers.

E.3.1 lqgarmdes.m

```
% lqgarmdes.m
%   TLA LQG Design
% Written by Dave Bossert
% Last Update - 10 Oct 94
%%% Load Synthesis Model %%%
load lqgsynth
%%%%% LQR Design for Gopt %%%%%
%%% Use only theta_1 and theta_2 inputs %%%
Blq = Bsynth(:,1:2);
Dlq = Dsynth(:,1:2);
alpha1 = 1;  %%% Theta_1 Tx Zero %%%
alpha2 = 1000;  %%% Theta_2 Tx Zero %%%
% Theta_1 Criterion
C1 = Csynth(1,:);% + alpha1*Csynth(3,:);
D1 = Dlq(1,:);% + alpha1*Dlq(3,:);
% Theta_2 Criterion
C2 = Csynth(2,:);% + alpha2*Csynth(4,:);
D2 = Dlq(2,:);% + alpha2*Dlq(4,:);
% Theta_1 Error
C3 = Csynth(5,:);
```

```
D3 = Dlq(5,:);
% Theta_2 Error
C4 = Csynth(6,:);
D4 = Dlq(6,:);
% Criterion Matrices %
Cyc = [C1; C2; C3; C4];
Dyc = [D1; D2; D3; D4];
% Define Weighting Criterion for Q
Q1 = .001;% penalty on Theta_1
Q2 = .002;       % penalty on Theta_2
Q3 = 50;         % penalty on Theta_1 Error
Q4 = 70; % penalty on Theta_2 Error
Q = [Q1 0 0 0; 0 Q2 0 0; 0 0 Q3 0; 0 0 0 Q4];
% Control Weighting (R), del_t1, del_t2
R1 = .001; % del_t1 penalty
R2 = .001; % del_t2 penalty
R = [R1 0; 0 R2];
% Determine Full-State Feedback Gain (G)
G = lqrcross(Asynth,Blq,Cyc,Dyc,Q,R);
% Partition G Matrix
Ge = G(1:2,1:18); % Estimator Gains
Gi = G(1:2,19:20); % Integral Gains
%%%%% LQG Estimator Design %%%%%
%%% Define Estimator State Space %%%
Ae = Asynth(1:18,1:18);
Be = Bsynth(1:18,1:2);
Gammae = Be;
Ce = [Csynth(3,1:18);Csynth(4,1:18)];
%%% Process Noise Matrix %%%
W1 =  20; %0.01; %20;   %%% del_t1 Noise (LTR)          %%%
```

```
W2 =  20; %0.01; %20;   %%% del_t2 Noise (LTR)          %%%
Wo = diag([W1,W2]);
%%% Sensor Noise Matrix %%%
V1 = 1;  %%% theta_1 noise  %%%
V2 = 1;  %%% theta_2 noise %%%
Vo = diag([V1,V2]);
Gff=[0 0;0 0];
%%% Optimal Gain, K %%%
K = lqe(Ae,Gammae,Ce,Wo,Vo);% lqgarmdes.m
%%% Combined LQG Closed Loop System %%%
Acont = [Ae-K*Ce-Be*Ge, -Be*Gi;zeros(2,20)];
H = [1 0;0 1];
I=eye(2);
Bcont = [K, Be*Gff; -H, I];
Ccont = -G;
Dcont = zeros(2,4);
%%% Evaluate Estimator & Regulator Poles %%%
damp(Ae-K*Ce);
damp(Asynth - Blq*G);
%%% Evaluate Command/Control Bandwidths %%%
% Input 3 - th_1cmd, Input 4 - th_2cmd
   %%% th_1/th_1_cmd %%%
Cx = [Csynth(3,:), zeros(1,18);Csynth(4,:),zeros(1,18)];
Bx = [Bsynth(1:18,:);zeros(18,4);zeros(2,2),-eye(2)];
Dx = [Dsynth(3,:);Dsynth(4,:)];
pbode(Acl,Bx,Cx,Dx,1,3,.01,100);
% Controller Reduction
Aest=[Ae-K*Ce-Be*Ge];
Best=[-Be*Gi,K,Be*Gff];
Cest=-Ge;
```

```
Dest=[-Gi,zeros(2,2),Gff];
%[ab,bb,cb,hksvl,T]=balmr(Acont,Bcont,Ccont,Dcont,3,12);
%hksvl
[Ab,Bb,Cb,Db]=canon(Aest,Best,Cest,Dest,'modal');
%[Ab,Bb,Cb,hsv,T]=balreal(Aest,Best,Cest);
%hsv;
elim=[1 2 3 4 5 6];
[Ared,Bred,Cred,Dred]=modred(Ab,Bb,Cb,Db,elim);
n=18-length(elim);
Acont2=[Ared,Bred(:,1:2);zeros(2,n+2)];
Bcont2=[Bred(:,3:9);-H,I];
Ccont2=[Cred,Dred(:,1:2)];
Dcont2=[Dred(:,3:9)];
  %%% Block Build Reduced LQG Closed Loop System %%%
  %%% Block 1: Synthesis Model (Plant) %%%
a1 = Ac;
b1 = Bc;
c1 = Cc;
d1 = Dc;

  %%% Block 2: LQR Optimal Gain %%%
a2 = Acont2;
b2 = Bcont2;
c2 = Ccont2;
d2 = Dcont2;

nblocks = 2;
blkbuild;
iu = [3 4];
iy = [1 2 3 6 11 12 13];
```

```
Q = [1,16,0,0
    2,17,0,0
    3,0,0,0
    4,0,0,0
    5,0,0,0
    6,0,0,0
    7,13,0,0
    8,8,0,0
    9,9,0,0
    10,10,0,0
    11,11,0,0
    12,0,0,0
    13,0,0,0];
[acl,bcl,C1,D1] = connect(a,b,c,d,Q,iu,iy);
```

E.3.2 bw.m

```
% bw.m
% Calculates Command BW for a plant and cascade controller (SANDY or PID)
% Written by Dave Bossert, Last Update - 10 Oct 94
load 18ordarmbs
a2=Abar;
b2=Bbar;
c2=[C(21,:);C(31,:)];
d2=[D(21,:);D(31,:)];
%SANDY
% a15good_c
% a1=A;
% b1=B;
% c1=C;
% d1=D;
% PID, 1,.5, 2 (extra pole at -10)
a1=[-10,0,0,0;1,0,0,0;0,0,-10,0;0,0,1,0];
b1=[1,0;0,0;0,1;0,0];
c1=[-199.5,5,0,0;0,0,-199.5,5];
d1=[21,0;0,21];
[aol,bol,col,dol]=series(a1,b1,c1,d1,a2,b2,c2,d2);
acl=(aol-bol*col);
pbode(acl,bol,col,dol,1,1,1,1000)
```

VITA

David Edward Bossert

Birthdate: October 11, 1962

Parents: Edward F. and Patricia A. Bossert

Advanced Degrees:

Bachelor of Science, Electrical Engineering,

University of Missouri-Columbia, 1984

Master of Science, Systems Management,

University of Southern California, 1987

Master of Science, Electrical Engineering,

Air Force Institute of Technology, 1989

Doctor of Philosophy, Aeronautics and Astronautics

University of Washington, 1996